PROCESS AND RELATION ANALYSIS
Capturing Architectural Thought

by

Paul E. Keel
Diplomierter Architekt HTL
Zentralschweizerisches Technikum Luzern
June 1992

Submitted to the Department of Architecture in partial fulfillment of the requirement for the degree of Master of Science in Architecture Studies at the Massachusetts Institute of Technology
February 1997

Signature of the Author

Paul Keel
Department of Architecture
January 17, 1997

Certified by

William L. Porter
Norman B. and Muriel Leventhal Professor of Architecture and Planning
Thesis Supervisor

Accepted by

Roy Strickland
Associate Professor of Architecture
Chairman, Department Committee on Graduate Students

Thesis Reader	**William J. Mitchell** Dean, School of Architecture and Planning Professor of Architecture and Media Arts and Sciences
Research Participant	**Charles M. Dalsass** Graduate Student, School of Engineering
Support	**Edith K. E. Ackermann** Visiting Lecturer, School of Architecture and Planning
	Thomas J. Allen Senior Associate Dean, SLOAN School of Management Howard Johnson Professor of Management
	Wolmer Facchin Research Assistant, ORL ETH Zurich Diplomierter Architekt ETH/SIA
	Gunter Henn Director, Henn Architekten Ingenieure, Munich Dr.-Ing., Architekt BDA
	Turid Horgen Research Associate, School of Architecture and Planning
	Christine Kohlert Architect, Henn Architekten Ingenieure, Munich
	Donald A. Schon Senior Lecturer, Urban Studies and Planning Ford Professor Emeritus
	Ralphe Wiggins Program Manager, Thinking Machines Corporation
	Susan Yee Graduate Student, School of Architecture and Planning

PROCESS AND RELATION ANALYSIS

Capturing Architectural Thought

by Paul E. Keel

Submitted to the Department of Architecture on January 17th, 1997 in Partial Fulfillment of the Requirement for the Degree of Master of Science in Architecture Studies

Abstract

This paper reflects the research conducted for the Design Technology Group at the Architectural Department at MIT under the participation of Charles Dalsass and the supervision of William Porter. The research evaluates electronic tools which support the formation of ideas in a collective design process. The tools focus on how to capture, analyze, and visualize concepts that develop from an individual or collaborative thought process. This work builds upon the user programming method developed by William Pena and its further development by Henn Architects in Munich.

William Pena's work introduces a shared knowledge base to enhance the user programming process. This shared knowledge base consists of cards pinned up on a board. The cards contain comments made during a meeting that can be viewed by all participants.

In this project, we investigate an electronic version of the previously described shared knowledge base. The electronic version provides advanced capabilities for remote collaboration, ease of storage, and manipulation of ideas. This builds the basis for follow-up explorations on how to relate, organize, visualize, and personalize the data contained in the knowledge base. Next, some corresponding methods will be developed to observe and visualize the concept formation process. The project will also discuss new ways to track the development process, the multiple use of the knowledge base for alternative purposes, and the synchronous and asynchronous manipulation of the knowledge base by remote participants. This study precedes the development of a computational solution, and therefore the last section of this paper will discuss the user interface and functionality of the proposed application. Although this research is centered around the architectural concept formation process, its content can be applied to various professions.

Thesis Supervisor William L. Porter
Norman B. and Muriel Leventhal Professor of Architecture and Planning

CONTENTS

INTRODUCTION

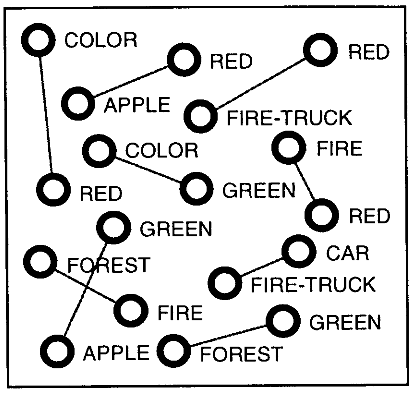
Advancements in information technology have influenced the manner in which we work, individually and in collaboration with others. New communication tools can permit designers to access global information and expertise which open opportunities of worldwide collaborative projects. As these information sources are more decentralized and abundant, the search for beneficial information becomes more complex and time consuming. Since work efficiency and speed of innovation largely depend on fast and accurate access to relevant information and structured communication among people, tools for information exchange and information analysis become of increasing importance.

The need for such information management and group collaboration is central to the following research, conducted for the Design Technology Group at the Department of Architecture at MIT under the supervision of Professor William Porter. The work builds upon the user programming method developed by William Pena (Pena, 1977) and its further development in the architectural practice of Henn Architects in Munich. This method provides a procedure for problem solving and idea development in groups by the introduction of a shared knowledge base utilized to enhance the user programming process. The physical artifact of this knowledge base consists of cards pinned on a board, called a *Card Wall*. The cards depict, through graphics and text, the comments made during meetings with programmers, clients, and users. The configuration of the cards on the wall, viewed by all the participants, shows the relationships of the various commentaries made during these brainstorming sessions. If we consider this card wall as an evolving object with which participants can capture, analyze, and visualize concepts developed from an individual or collaborative thought process, then we may begin to conceive a set of electronic tools which facilitates the formation of a Card Wall, improve its efficiency, as well as provide additional capabilities specifically afforded by computation which help to support the development and understanding of ideas during a collective design process.

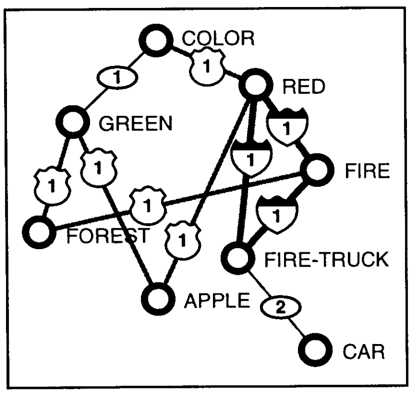
This paper reflects our investigation of an electronic version of the shared knowledge base inherent in the card wall. This computational version may provide advanced capabilities for remote collaboration, ease of storage, and manipulation of ideas, which build the basis for some explorations on the relation, organization, visualization, and personalization of the information within the knowledge base. The project also examines new ways to track the development process, the multiple use of the knowledge base for alternative purposes, and the synchronous and asynchronous manipulation of the knowledge base by remote participants. The last section of this paper will discuss the user interface and functionality of the proposed computational solution. Although this research is centered around the architectural practice, its contents may be applied to other disciplines.

DESIGN OPERATIONS

As a prelude to the description of the physical and electronic card wall and its specifications and features, this section reviews some assumptions of mental perception as it relates to the design process. It also serves as an introduction of the terminology used in subsequent sections of this text.



Information Fragments Figure 1



Networked Information Figure 2

Since designers and engineers accumulate knowledge while working on a task, we will refer to the *design process* as a *problem solving and learning process*. Learning allows for the accumulation of new knowledge as well as the understanding and resolution of a particular problem. We will consider the learning process as combination of *recalling (storing and retrieving) and understanding (relating)* accumulated knowledge. For example, we may recall the notion of a "color red" and a "red apple" or, in addition, recognize a possible relation between two "red" things. By linking related information it is possible to reduce redundancy and storage, as well as providing recognition of relationships which may provoke innovations in cognitive processes. Figure 1 and 2 illustrate a more complex example with many fragments of information pieces that become related.

In the present stage of information technology, the computer often represents a very powerful tool to store enormous amounts of *information*. However, there are very limited capabilities to relate and interpret information into a relevant knowledge base. This research emphasizes the use of the computer as a tool to store and retrieve information as well as one which organizes and visualizes relationships between information. The proposed computational tools seek to support the problem solving process of individuals and groups.

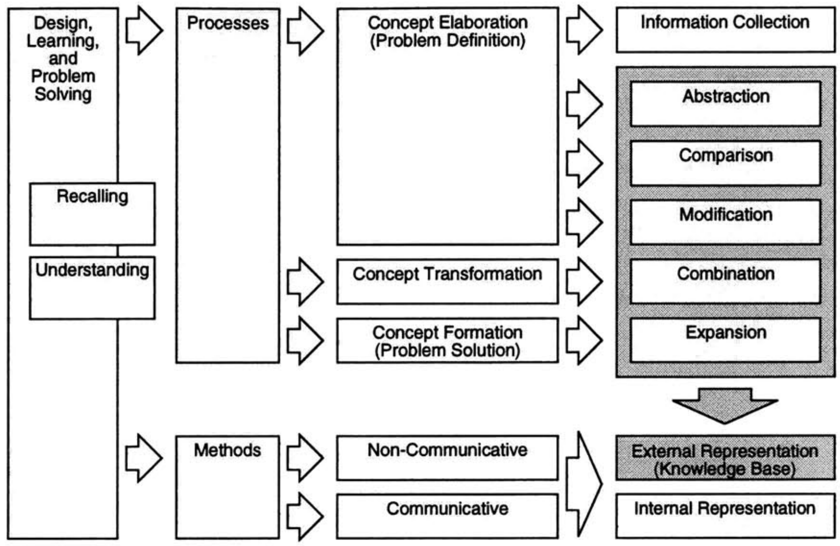
Figure 3 attempts to formalize the problem solving activity by its division into *processes and methods*. Processes refer to the various subactivities while the word "method" is used to specify different forms of knowledge accumulation during a design activity.

Processes

For an easier understanding of the later part of this research we will divide the design process into three stages: concept elaboration, concept transformation, and concept formation. The performance of these stages is not necessarily sequential, but often iterative. This separation of design processes is important for our consideration of computational tools that are flexible enough to be used at any stage as well as for those facilitating easy transitions between design stages.

Concept Elaboration (Problem Definition)

In the concept elaboration phase we define the design problem rather than the design solution. The design problem is discovered by the initial definition of the design goal, the comparison of predefined concepts (solutions to common problems), and the careful collection and analysis of related information. The analysis process often requires the abstraction of information to permit easy comparison and modification. Abstracting information is achieved by extracting and simplifying relevant data. One approach to design abstraction is the use of diagrams which capture and emphasize the essence of a particular idea. The entire concept elaboration phase is often referred as brainstorming. If information is specifically obtained from the client, we refer to it as user programming.

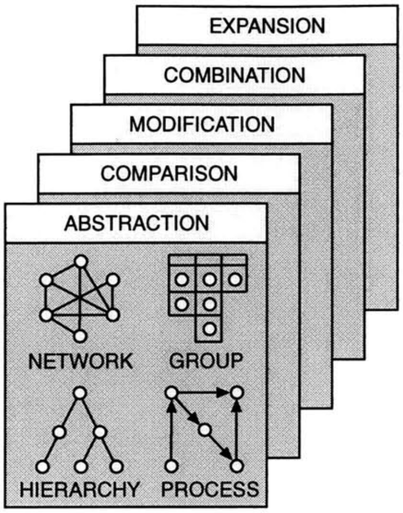


Design Processes and Methods Figure 3

Concept Transformation

The concept transformation can be considered the transition between the problem definition (concept elaboration) and problem solution (concept formation) phase. The information collected during the concept elaboration phase is combined, structured, and organized during the concept transformation phase. This is a form of *sense making*. Because the individuals involved in the concept elaboration phase (e.g. user programmers) are sometimes not the same as those involved in the concept formation phase (e.g. designers), it is necessary to effectively and efficiently transfer information collected during the concept elaboration phase to those people involved in the concept formation stage. A typical product of this transfer is the *brown sheet* introduced by William Pena and the commonly used *master plan*, which will be discussed later.

Diagrams Figure 4



Concept Formation

The concept formation phase can be seen as a further refinement of the data organized during the concept transformation phase. The actual solution to the problem is developed at this stage of the design process. The development from concept elaboration to concept transformation to concept formation embed various sub-activities such as *information collection, abstraction, comparison, modification, combination, and expansion*. The transition between the five conceptual stages in figure 4 is achieved through an abstract handling of data. The Diagrams allow for the visualization of data on this abstract level. This research focuses on diagrammatic representations as a tool to isolate, organize, connect, and store ideas. We concentrate on an elementary type of diagram consisting of *nodes* and *connections* between nodes. We will use nodes to symbolize data elements and connections between nodes to express relations between data elements. Diagrams not only facilitate the transition between abstraction, comparison, modification, combination, and expansion but also allow for some shared understanding of the data by the extraction of core elements or ideas. Data expressed in this abstract form may allow the efficient comparison of present and past design solutions.

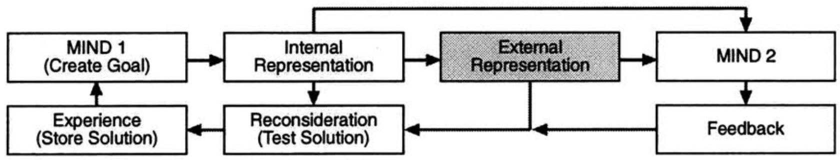
Methods

Problem solving methods refer to the different forms of knowledge accumulation. We differentiate between knowledge that is accumulated 1) with or without the use of tools and 2) with or without the participation of other individuals. A simple mathematical expression, for example, can be solved *internally* without the use of an external representation or the help of another person (compare figure 7). This is achieved by initially *establishing a goal* (in this case solving the expression), *constructing an internal representation* (computing a result), *reconsidering the outcome* (testing if the result is possible), and finally *drawing conclusions from this experience* (accumulating knowledge). We repeat this loop until the final result becomes satisfactory to us.

An internal representation is limited by our memory capacity. Usually we work on one sentence, one part of a mathematical problem, or one detail of a design at a time. The handling of more complex issues requires the extension of the loop by an *external representation* such as a piece of paper. Our internal representations are supported by external representations or tools. The efficient use of external representations allows us to unburden our mental database and to solve more complex tasks. This disembodiment also helps us to reconsider solutions.

Jean Piaget explains *reconsideration* as the basis to create experience and construct knowledge. Edith Ackermann suggests that the main advantage of external representations is to make explicit one's own process and to be aware of one's own activities. She further states that because of the use of external representations we experience various improvements concerning 1) the evocation of an object at another time or place, 2) the ability to keep track of the past, 3) the way to describe one's activity, 4) the ability to take a more complex detour, 5) the control of one's activity, and 6) the decision of whether to act or not to act.

Another method to solve a problem or to develop an idea is to discuss the task with others. We verbally explain the present stage of our internal representation to another person or use an external representation to transfer our thoughts as accurately as possible to the other persons involved. This usually results in some sort of *feedback*. Feedback is processed (reconsidered) in the same way as internal and external representations. This ongoing loop of reconsidering internal and external representations as well as feedback provided by others may be considered central to our ability to solve problems and develop new ideas.



Internal/External Representations Figure 5

Figure 5 illustrates four methods to solve a problem or develop an idea:

1) one person develops an idea internally
2) one person develops an idea internally with the use of external representations
3) more than one person develop an idea together through verbal interaction
4) more than one person develop an idea together through verbal interaction in conjunction with the use of external representations

Knowledge Base

Since the shift from the industrial age to the information age we find ourselves in a world of increasing complexity that suffers from an overload of information. Tools to organize and visualize information are not only important for the development of ideas but for the accessibility and understandability of information in general. This research will address this issue in regards to the redesign of the Card Wall.

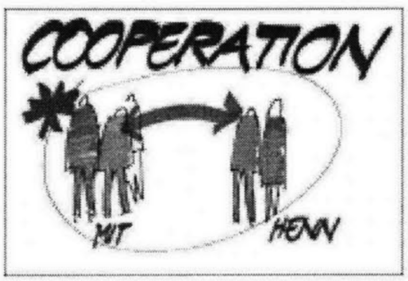
An external representation can be seen as a thought frozen in time that can be analyzed and retrieved more easily. It can also be seen as a visualization of an idea which may lead to new ideas. Many external representations might build some sort of idea pool or *knowledge base* that is accessible to many, rather than one person. The Card Wall is an example of such a knowledge base (the cards contain ideas and the wall illustrates relationships among ideas). Relational information is visualized by assigning the cards to groups (headings). Therefore, the participants of a group discussion not only share the content of cards but also some understanding about the relations among cards. We will later explore additional possibilities on how to enhance the visualization of relations among cards.

A *shared knowledge base* is a network of data assembled by many people of diverse backgrounds at different locations and times. The participants involved in the creation of a shared knowledge base share a common interest or participate on the same project. This model of group collaboration usually leads to more innovation and better coordination among the members of a team. Since we consider the Card Wall to be a shared knowledge base we will reflect on issues that concern user interaction and group collaboration.

Since the cards on the Card Wall represent pieces of information, the relational connections among cards visualize the present stage of the development. This network of relationships may also contain conclusions drawn from the information collected. In linguistic terms we would speak of the cards as the *vocabulary elements* and of the links between the cards as the *grammatical rules*. A few heavily linked cards for example might result in a restricted set of possible conclusions (small vocabulary with many rules). Both, vocabulary elements and grammatical rules can be stored, organized, and visualized. We will explore this theory in more detail.

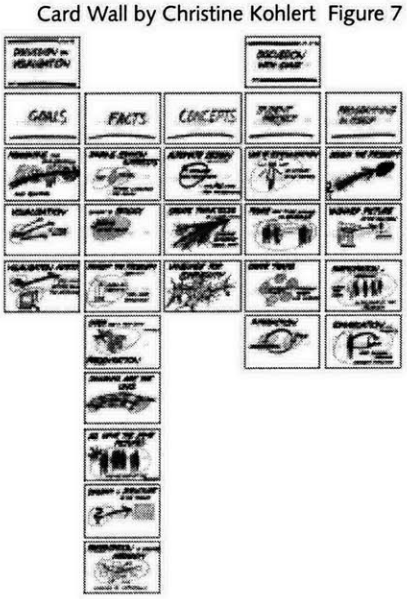
THE PHYSICAL CARD WALL

Card by Christine Kohlert Figure 6



The Card Wall can be seen as a shared knowledge base that is created and accessible by many rather than only one person (knowledge collaboration). The relationships between the cards on a Card Wall represent the definition or the solution to a problem. The main idea is to collect and display abstract ideas from groups during brainstorming sessions in a systematic way. During group meeting comments, suggestions, or ideas are graphically "written" on individual cards and are pinned up on a wall. The wall becomes a collection of cards that can be viewed and compared by all participants. Often the cards are organized on a grid and categorized under pre-defined topics. The x-axis separates cards by goals, facts, concepts, needs, and problems while the y-axis divides function, form, economy, and time.

Card Wall by Christine Kohlert Figure 7



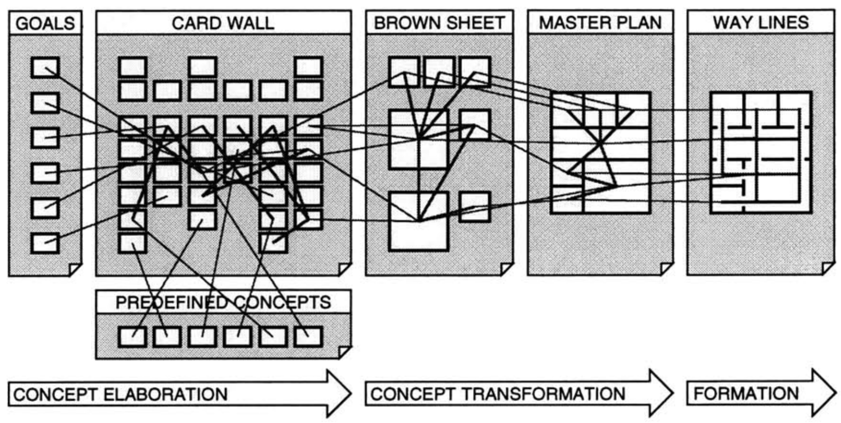
An advantage of the physical Card Wall is that knowledge remains accessible and can be used for future projects. Participants can easily switch between the various issues which have been discussed. Since the Card Wall is organized in only one way, the location of cards can easily be recalled. The Card Wall allows for the review of a meeting by non-participants or members working asynchronously. It also supports the preparation of post-meeting sessions.

A disadvantage of the physical Card Wall is that there is a certain limit to the amount of cards that can be related mentally. Some relationships are changing over time and are difficult to understand and organize as they change. In addition, past stages of card walls are difficult to compare or organize. Because of the static nature of the Card Wall, the cards can not be rearranged and relations among cards are difficult to visualize. Finally, the size of the Card Wall limits its duplication and transportation.

ELECTRONIC CARD WALL APPLICATIONS

A computational version of the Card Wall may be able to maintain the advantages of the physical version as well as providing solutions to overcome its disadvantages. It should not only support the designers work through various stages, but also should allow for an easy transition between the different activities of the design process. The ideal solution will also support the transfer of a project from one work-group to another during the design process keeping the program open and flexible for future additions. The use of the traditional Card Wall is largely confined to support the problem definition stage. This section introduces the electronic Card Wall as a tool to be used in other parts of the design process already described; that is, concept elaboration, concept transformation, and concept formation.

Card Wall Applications Figure 8



Goals

From the previous sections we know that a solution to a problem proceeds from the definition of the problem and the definition of a problem stems from some definitions of goals. Goals may be considered the objectives to be achieved during the problem definition process. The satisfactory completion of the task defined by the previously proposed goals approve the transition from the concept elaboration to the concept transformation phase. The initial definition of goals is also helpful for possible considerations of subdividing the task. Goals can be represented by and related to the cards on the Card Wall. Heavily linked goal cards indicate their careful consideration during the problem definition phase.

Card Wall

The electronic version of the Card Wall is discussed later and provides the ability to link and move cards with their links attached.

Predefined Concepts

Predefined concepts are repetitively used concepts or combinations of concepts from previously developed Card Wall's. They are based on the designers design philosophy or past design experiences. They are viewed as cards or links between cards. An example of a concept is the word "decentralization" in management or "communicative space" in architecture. An example of a concept combination is a relation previously made between the two concepts "user collaboration" and "communicative space". An electronic version of the Card Wall would allow for the recognition of predefined concepts and the exchange of predefined concepts among organizations.

Brown Sheet

An outcome of the traditional Card Wall as practiced by Henn Architects is the generation of a brown sheet and a master plan, which are used to transfer relevant knowledge from the Card Wall to the concept formation stage. The brown sheet introduced by William Pena transforms the data collected during an architectural user programming session into a *list of space requirements for a building*. The Brown Sheet is essentially a large sheet of brown paper with square white boxes that represent the space required for each room in a building. The sizes of the boxes indicate an approximation of the space necessary for each room. The objective is to decide how many rooms can be realized and how much space can be allocated for each of the rooms.

An electronic version of such a tool would allow for the automatic generation of box sizes based on the proposed amount of area for each room, as well as the indication of the total space used by all rooms. Boxes could be linked to visualize the desired connections among rooms. Each of the boxes could also be linked to the cards that provoked the decision about size and location. Since the electronic Card Wall would allow moving and linking boxes, the electronic Brown Sheet requires only a few additional functions such as the ability to resize and generate boxes based on numeric input.

Master Plan

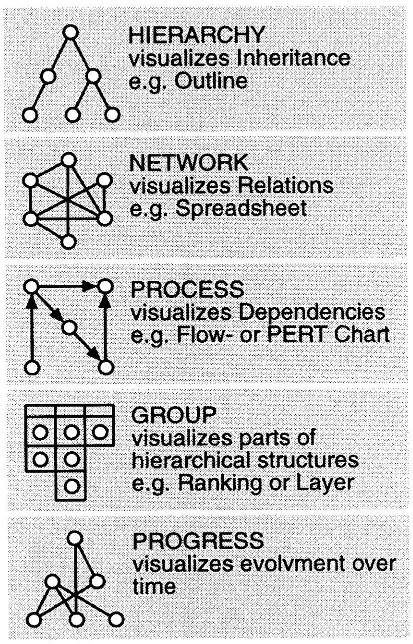
The master plan is a *visualization of a spatial room arrangement*. The room sizes are adjusted to make all rooms fit into a confining space. An electronic Master Plan would allow designers to arrange and adjust rooms directly on the electronic Brown Sheet. The electronic Master Plan would not require any additional functionality since the electronic Brown Sheet already allows boxes to be moved and re-sized.

Way Lines

In this preliminary design stage of the actual building artifact, one of the many tools which may be adopted is our notion of Way Lines. Way Lines build the bridge between the Master Plan and the actual design of the building. Way Lines are considered to *support the arrangement of doors, windows, and furniture*. The room connection lines from the Brown Sheet and the spatial room organization from the Master Plan are used as guidelines for Way Lines. Way Lines are based on the same functionality as the electronic Card Wall, Brown Sheet, and Master Plan (resizing and moving boxes). Way Lines therefore, provide an additional tool that can easily complement the electronic Card Wall. Way Lines are the topic of another research and are beyond the scope of this paper.

CARD CONNECTIONS

We have mentioned that the most abstract form of a diagrammatic representation is an assembly of nodes (cards) and connections between nodes (links). In relation to the Card Wall, we will distinguish among five different types of possible connections. Each connection type serves a different purpose for discrete and specialized organizational purposes.



Card Connections Figure 9

Hierarchy

The hierarchical structure is the most common type of data organization. Hierarchical structures are often utilized to represent organizational structures, text outlines, or assemblies of mechanical parts. Hierarchical structures always require a top object to facilitate the separation and isolation of specific trees of related objects.

Network

Network links express relations among objects without implying precedence or direction. For example, a network connection is the relation between the two words "art" and "painting". Typical examples of networks are "spread sheets" that relate objects along the x- and y-axis. Since a network link does not group or rank objects according to previously defined headings it allows for a rapid and less structured linking of objects.

Process

The process link expresses dependencies between processes. Consider the two words "cooking" and "eating". There is obviously a relation between the two words. However, since both words represent part of a process and since the cooking process precedes the eating process, we recognize some sort of dependency among the two activities. Unlike other links, the process link uses arrows that allows for the expression of direction. Processes are typically used in project management and scheduling.

Group

Many cards attached to one heading build a group. A group can be considered a single level hierarchy. The main difference between a group and a hierarchy is that a group object can belong to more than one group heading. This might make it look like a network. However, since a network has no official top (group heading) we will consider the group links as something unique. Engineering drawing software such as AutoCad typically represent groups as layers.

Progress

The progression link visualizes the chronological order of cards. Similar to the process link, the progression link is directional. Since progression is usually visualized from the top to the bottom and from the left to the right, arrows to indicate direction are usually not necessary. Illustrative examples of progression links are found in calendar scheduling programs.

CARD WALL CONCEPTS

A computational rendering of the Card Wall opens opportunities of using it and thinking about user programming in different ways. This section describes four possible re-interpretations of the traditional card wall afforded by the electronic version. They are the individual card wall, the non-dimensional card wall, the democratic card wall, and the unified card wall.

The Individual Card Wall

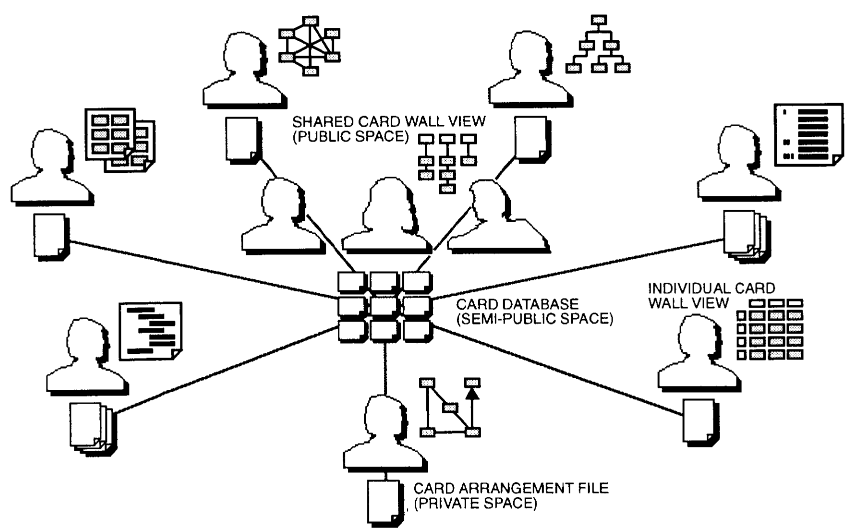
If we consider the Card Wall as a shared knowledge base allowing a shared understanding of its content, then we are assuming a match between the card wall as a external representation and a shared internal representation. This section questions this compatibility of the traditional Card Wall with the internal representations of each group member. An electronic Card Wall may easily extract and layer individual interpretations making them explicit to others and ready for further discussion.

Jean Bamberger did significant work on children's comprehensibility. In one of her presentations at the Media Lab in early November she concluded with two important assumptions:

1) *The description on a piece of paper reflects the internal representation one is using.*
2) *One's actions are meaningful to oneself.*

The participants of her presentation were encouraged to test these assumptions through the making of an individual external representation of a piece of music with graphical symbols. Almost all participants constructed different solutions representing the same piece of music. Only a few people were able to understand the solutions created by other participants.

Remote Card Wall Access Figure 10



This test leads us to the assumption that people might also perceive the information contained on the Card Wall in different ways. Even if participants share the same content they do not necessarily share the same interest or focus. Participants are likely to have diverse views of the Card Wall and each participant would probably group the cards in a different way. Since shared information is contained in both, the cards and the location of cards, we need to consider possibilities that allow for both shared and individual Card Wall arrangements. The match between the participants shared external and individual internal representation is likely to enhance user participation and inspiration.

Figure 10 illustrates a group of people working on a Card Wall. The center of the picture displays a conventional Card Wall with people sharing the same Card Wall view. The individuals off the center participate physically or remotely and view the contents of the Card Wall in different arrangements on their personal screens. The shared Card Wall view may always be referenced at any time. The *University of Arizona* developed a similar, text based, system called *EMS (Electronic Meeting System)*. Participants of an electronic meeting were arranged around a table with personal computer terminals. Each of the participants could decide independently what to contribute and enter comments directly through his terminal. The comments were organized by a discussion leader and displayed on a big screen. The remarks could also be rearranged individually by each participant on his personal terminal. This system allowed participants to keep track of comments made by others, insert anonymous comments, and secure equal opportunities for participation.

The main advantage of having many people viewing only one Card Wall arrangement (as practiced by Henn Architects) is the shared perception of card locations and evolving card patterns. Since the cards on the Card Wall are grouped by topics, the participants also share some understanding about the relations among cards. However, the recognition of relations among cards becomes increasingly difficult as the Card Wall grows. This makes the rearrangement of cards a necessary but formidable task. Henn Architects rearrange cards through the creation of *posters and brochures*. A poster is a visualization of a selection of related cards on a big sheet of paper. A brochure is a little booklet that illustrates the various stages of the Card Wall and defines the meaning of the individual cards. Posters and brochures are usually created after a meeting and helpful for the preparation of post-meeting sessions. Like posters, the possibility for individual arrangements allows people to group a few cards in a understandable way which complements the Card Wall.

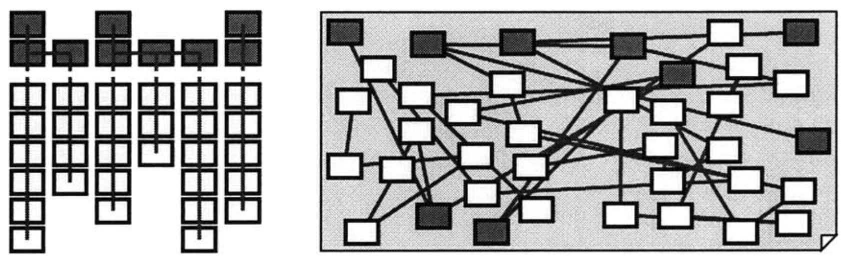
The Non-Dimensional Card Wall

This view of the Card Wall questions the advantages of fixed card locations which have been mentioned previously. Data is traditionally organized in a hierarchical manner and visualized in two-dimensional space. Today's computer technology might allow a different and more sophisticated way of organizing data. Cards and relations among cards on a Card Wall that are completely autonomous and detached from their physical locations would not only allow for more flexibility, but may encourage new ways of handling and viewing information.

The left half of Figure 11 illustrates a traditional Card Wall layout. This layout represents the equivalent of the randomly dispersed , but properly linked cards on the right. We can visualize relations among cards by either grouping them in a traditional Card Wall layout or by linking them accordingly. The computer is capable of generating the links between cards based on a traditional Card Wall arrangement. In addition the computer may generate a Card Wall arrangement based on the links between cards. Relations among cards in represented in terms of links rather than by means of physical card locations not only allows for the automatic generation of Card Wall arrangement, but may generate other card arrangements as well. In the following parts of this thesis, we will discover other card arrangements and manipulations.

A major advantage of working with links rather than card locations is that cards and links can be exchanged among people that view cards in different arrangements and with different tools. We will refer to this type of organization as a non-dimensional space since it depends on links among cards rather that card locations in space.

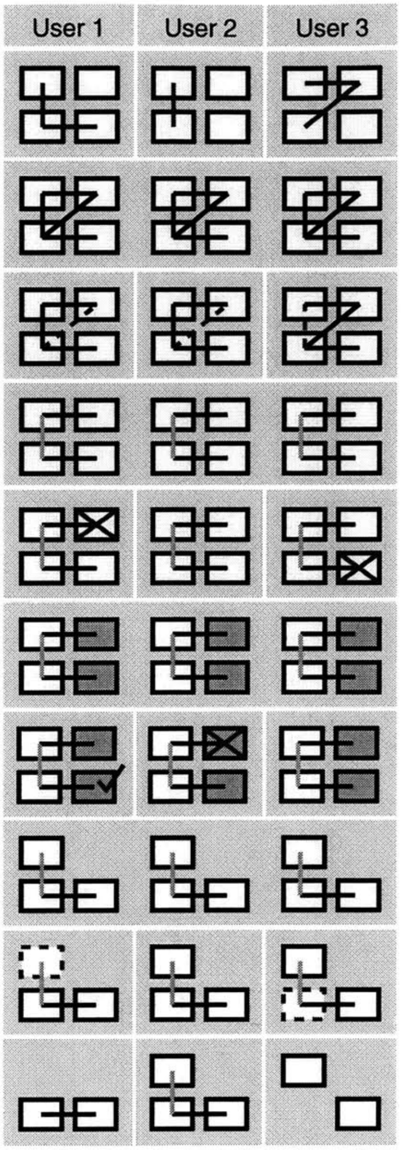
Non-Dimensional Card Wall Figure 11



The Democratic Card Wall

When user programming is practiced with the traditional Card Wall, non-team members create cards and pin them up. These people are specifically trained to select from the comments made by team members. In addition to this, they to interpret the comments, represent comments in written and graphical form on cards, and arrange cards on the board. The fact, that the decision about the importance of an idea and its location on the Card Wall is made by a non-team member might lead to some concern about accuracy of the Card Wall. However, since those people share the table with the team-members, misunderstanding's are usually recognized and corrected.

The democratic Card Wall allows for remote participation and decision-making in group discussions. It empowers team-members to add and subtract information on a Card Wall not only during but also before and after a meeting as well as to discuss issues in parallel. However, the difficulties with the democratic card Wall are best explained by an example. Figure 12 illustrates the operations made by three users working remotely on a Card Wall.



Democratic Card Wall Figure 12

Row 1 Presently, the Card Wall contains four cards. User 1 decides to add two links. User 2 decides to add one link. User 3 decides to add two links.

Row 2 All users see all the links created by all the users.

Row 3 User 1 and 2 do not agree with the diagonal link. User 3 does not agree with the link created by user 1 and 2.

Row 4 The diagonal link disappears since only one out of three users created it and two out of three users voted against it. The link created by user 1 and 2 changes in color to indicate that not all participants agree on this link and that only a few more votes are needed to make it disappear completely.

Row 5 User 1 decides to erase the card on the top right and user 3 decides to erase the card on the bottom right.

Row 6 The cards erased by user 1 and 3 change in color to indicated the disagreement on this card.

Row 7 User 1 decides against user 3 and check-marks the card on the bottom right. User 2 agrees with user 1 and also decides that the card on the top right should be erased.

Row 8 The card on the top right disappeared since two out of three users voted against it. The card on the bottom right changes back to its original color since only one user disagrees and one user agrees on this card.

Row 9 User 1 and 3 erase one of the cards in their local work space to obtain an individual view of the Card Wall.

Row 10 All users view the Card Wall differently. The Card Wall still contains the common information and allows users to restore the shared view at any time.

The democratic Card Wall implies various difficulties and disadvantages. Valuable comments developed by a minority can easily be eliminated by a majority that does not yet realize the potential of the former's observation. The decision making process would obviously slow down. The management for such a system is very complex and the changes made by users are difficult to control.

The advantages of the democratic Card Wall imply awareness of disagreement, the ability for parallel and remote collaboration, anonymous participation, and the advanced influence of team-members in the Card Wall generation. The computer can keep a record of all additions and deletions or visualize the tally of the votes for cards and links, perhaps with a bar graph on the card border or by changing the color of links.

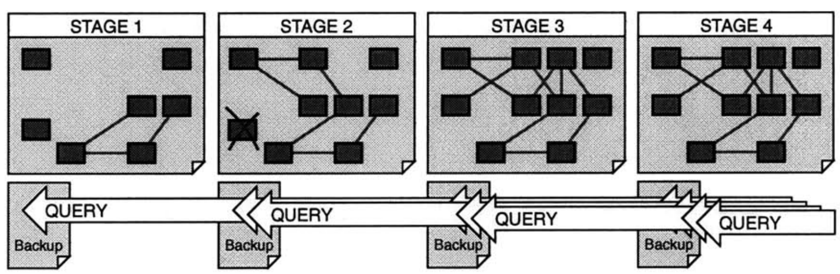
The Unified Card Wall

This chapter discusses ways to record the progression of the generation of the Card Wall. We will initially analyze the conventional methods used and later explore alternatives that not only allow for an easier and more flexible handling of the Card Wall but also encourages new ways of thinking and collaborating.

Divergent Process

The brochures produced by Henn Architects capture various stages of the Card Wall generation process. We will call this frequent backing-up of Card Wall stages a divergent process. A divergent process allows for the comparison of previous and present versions of the Card Wall generation process as well as for the comparison of Card Wall versions that were created by different teams. The maintenance of prior of Card Wall stages becomes important if major changes are made to it. For example if the cards on a Card Wall are rearranged, it is important to be able to go back to a stage prior to rearranging the cards. The ease of recognition of differences between two Card Wall versions depends on the amount of cards on the wall. An electronic Card Wall could possibly provide a slider that allows users to view the Card Wall development over time. The complexity of comparisons also increases with the amount of saved Card Wall versions. Figure 13 illustrates four saved versions of a Card Wall that result in ten possible comparisons. The divergent process requires the careful selection among comparisons of Card Wall versions and, if possible, the visualization of the differences among the Card Wall versions.

Divergent Process Figure 13

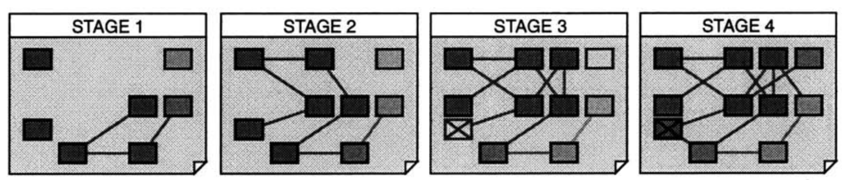


Unified Process

In practice, we are unlikely to compare past versions of a development process since this task is time consuming and often leads to confusion. Valuable information often gets lost in previous versions of a development process. Every rearrangement of the Card Wall contains the risk of information loss if the present stage of the Card Wall is not saved first. Having to save a Card Wall stage before every manipulation obviously results in too many saved versions which are almost impossible to manage and compare. We will therefore imagine an alternative possibility that allows for the simultaneous visualization of the present and the previous versions of the Card Wall.

The unified Card Wall keeps track of Card Wall generation and visualizes it as an integral part of any Card Wall arrangement. Figure 14 illustrates this idea. 1) The first generation of the unified Card Wall contains a few cards - some of which are linked together. We assume that the linked cards have something in common or form some sort of relational concept. The two cards on the left have been recently created and are not yet linked. The card on the top right was created earlier and is automatically grayed out (weakened) since it has not been linked to any of the other cards for a long time. The program assumes that the user did not link this card to the other cards because he either forgot about this card or the card has nothing in common with any of the other cards. The weakened card reminds the user to reconsider its value as well as its possible relation with other cards. 2) During the second generation, the user links the two cards on the left with the other cards on the wall and still does not consider the card on the top right. Therefore, the program continues to weaken this card. The program also starts weakening the middle right card since the only link attached to it had been created a long time ago. 3) During the third generation the user does not notice the weakened cards. Instead, the user decides to erase the card on the bottom left. Rather than erasing the card entirely, the program indicates the new status of the card by crossing it out and weakening it by the maximum amount possible. The computer also has decided to weaken some of the other cards since some of them have only a few links attached or are linked in a chain. Cards linked in a chain (linear) often relate similar concepts that substitute each other (we will explore this theory later in this text). 4) In the last generation of our example the user finally becomes aware of the weakened cards and reexamines how they belong to the other cards by linking them appropriately. The user also decides to attach an additional link to the card that he previously erased. This strengthens the card but the cross still indicates that it is officially non-existing.

Unified Process Figure 14

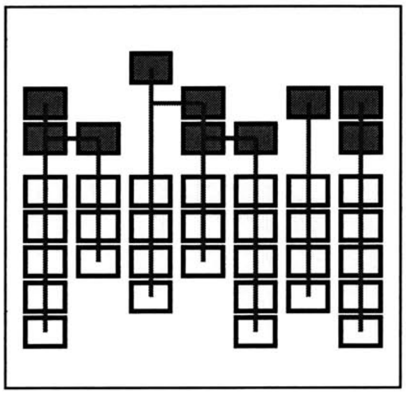


The previous example illustrates only one way of controlling the Card Wall development. Tests need to be conducted to find out about intelligent control mechanisms that optimize the functionality of the Unified Card Wall. The advantages of the Unified Card Wall are that (1) forgotten cards act as a reminder of ideas which may still hold value, (2) cards can be eliminated in a gradual way, (3) no previous Card Wall's need to be compared, (4) information does not get lost, and (5) cards that are irrelevant to the task get automatically excluded. The Unified Card Wall also supports the management of the previously proposed Democratic and Non-Dimensional Card Wall.

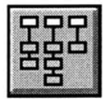
The Unified Card Wall can be compared to the World Wide Web. WWW sites represent pieces of information that are linked to other pieces of information. Heavily linked sites represent obviously the more interesting or popular pieces of information. Sites of decreasing value lose links and at some point become entirely detached. Both the Unified Card Wall and the WWW represent a constantly updated information pool in which only valuable pieces of information survive.

CARD WALL ARRANGEMENTS

This chapter explores 12 different Card Wall arrangements that allow enhanced data visualization and the recognition of evolving data patterns. Constant switching between various arrangements might allow for a partial match with our internal representations. Some of the arrangements were developed for the specific use with certain *card connection types (Hierarchy, Network, Process, Group, Progress)*. However, all link types as well as combinations of link types can be viewed with each arrangement.

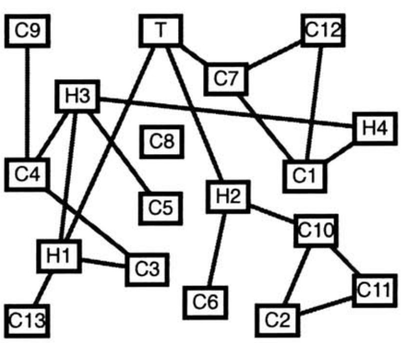


Card Wall Arrangement Figure 15

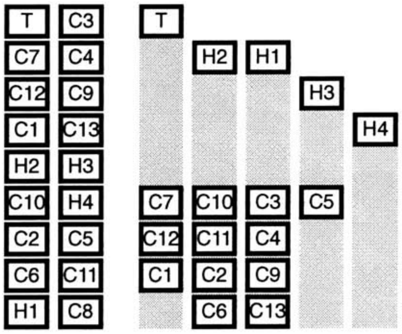


Card Wall (Hierarchy)

The Card Wall view is one of the most important arrangements for Henn Architects. It is the primary organizational method used by Henn during their programming sessions. This arrangement provides middle ground between the traditional Card Wall use and its computerized and automated use which this thesis proposes. It allows users to work with the Electronic Card Wall in a conventional and familiar way. The traditional Card Wall forces the hierarchical structure of data and does not allow for links among cards between groups. A flexible solution demands the automatic generation of Card Wall arrangements from linked but randomly dispersed cards (non-dimensional space). In addition, the manual generation of a Card Wall arrangement on a conventional two-dimensional workspace is needed to complete the solution. These two solutions are described below.



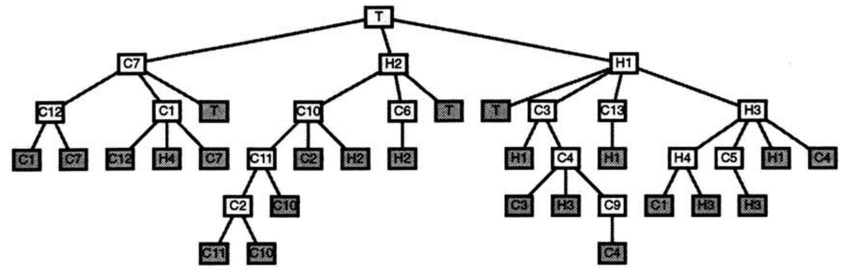
Netwoked Information Figure 16



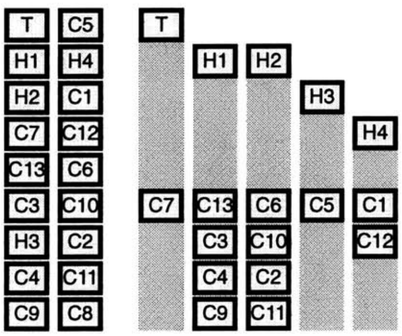
Clockwise Execution Figure 17

Non-Dimensional Setting

In a non-dimensional setting, the computer checks for hierarchical links among cards. This requires the selection of the hierarchical top by the user. Many Card Wall solutions exist if cards are not specifically linked hierarchically. In fact, hierarchical order can be created from a set of highly random cards and links. Consider the following example: Figure 16 illustrates randomly linked cards (C) and headings (H). This assembly of cards is reorganized in a Card Wall arrangement under the application of a *recursive function* (a recursive function is a computer function that can call and execute itself repeatedly). The recursive function initially takes the pre-defined top card and places it on the top left corner on the Card Wall area. The function then examines the card clockwise for its attached links. This makes C7 the next card to undergo analysis. The recursive function places C7 below the top on the Card Wall area. (A card can be below a heading even if this heading has sub-headings. This happens if the user did not yet decide to what sub-heading this card belongs or if a new sub-heading should be created.) The next card examined is the heading card H2. (Heading cards are ignored if the source of the link is not a heading.) In our case, the source of the heading card H2 (also called the calling card) is the pre-defined top heading. The heading H2 is placed on the heading area on the Card Wall. Each heading is placed to the right of the previously placed heading and below the calling card. The recursive function proceeds with the analysis of H1 and continues to examine the links of C7, H2, and H1 accordingly. Figure 18 visualizes the recursive execution as a tree. The dark boxes indicate cards or headings that were previously placed and consequently have not been considered again. The recursive function finishes its execution when all leaves point to previously placed cards or headings.

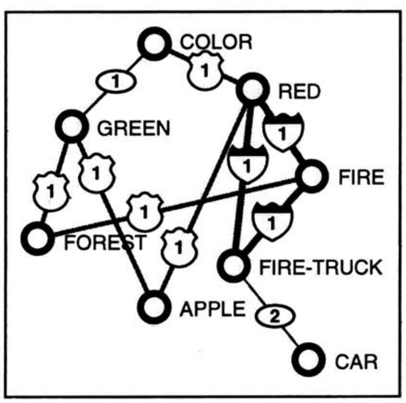


Recursive Function Figure 18



Counterclockwise Execution Figure 19

The card arrangement on the Card Wall changes if the function is executed counter-clockwise (compare figure 19) or if a change is made to critical links in the structure. Users of this Card Wall would obviously become confused if cards appear in different positions every time the Card Wall arrangement is regenerated. One possible solution to this is to make the clear *distinction between hierarchical links and network links*. This forces the user to specify the type of link he is creating. Hierarchical links could automatically be tested for correctness. This solution would allow a clear reconstruction of the Card Wall. However, it does not yet provide a satisfactory solution to the problem, since it is obviously a difficult task to hierarchically link dispersed cards on a board. It is also important that users concentrate on the subject rather than what types of links they are using.



Networked Information Figure 2

Assuming that no hierarchical structure among the links could be identified, there is another alternative. It is possible to *temporarily duplicate the cards that are assigned to more than one heading*. This allows users to link one card to more than one heading. The Card Wall arrangement would display a temporary copy of this card under all headings it belongs to and users would not have to worry about hierarchical links. This might present an ideal solution to the problem. However, a user might accidentally create a *link loop*. A link loop emerges if for example H1 is linked to H2, H2 to H3, and H3 to H1. This results in infinite duplications of cards on a Card Wall arrangement. There would have to be some sort of logic within the function to prevent this situation from happening.

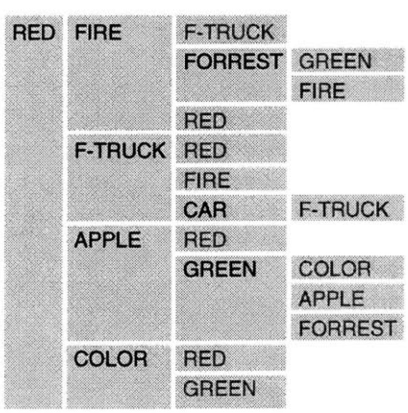


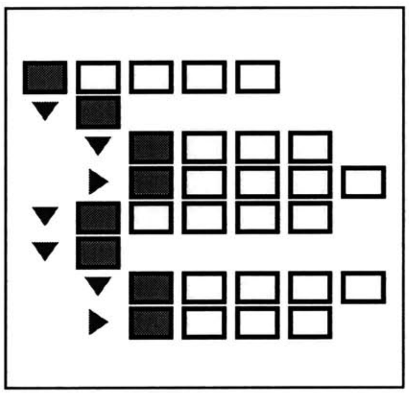
Figure 20
Hierarchical Visualization of
Networked Information

The previous two paragraphs indicate that a Card Wall arrangement consisting of non-hierarchically linked cards may be generated in two possible ways: 1) Cards that belong to more than one heading are only listed once. This implies that if the link structure changes, the cards might change their position as well. 2) Temporary copies of cards can be displayed under all headings the card belongs to. In this case there is a possibility that the amount of duplications can become infinite.

We will now consider a third alternative that does not require a separation between hierarchical and network links. This method also allows for the duplication of cards that are assigned to more than one heading. Figure 2 illustrates a network with many link loops and figure 20 shows the hierarchical tree derived from the word "red". The *construction rule* requires the clockwise link analysis. All words attached to the word "red" create the next level of sub-headings that are analyzed accordingly. The *termination rule* says that if sub-headings exists already, do not use it twice in the hierarchy.. The word "apple" for example is a subheading of "red". "Red" and "green" belong to the sub-heading group of "apple". The sub-heading group of "red" is already defined and needs no further explanation. The sub-heading group of "green" is not yet defined and therefore produces a sub-heading group that consists of "apple", "forest", and "color". This method allows the hierarchical definition of networked nodes by avoiding infinite recursion.

Two-Dimensional Setting

In a two-dimensional setting, the user arranges the cards according to the traditional Card Wall layout by dragging or moving the cards one at a time. The electronic Card Wall version supports this action with a snap mechanism that automatically fits the cards on a grid. When the cards are in the appropriate order an *automatic link generator* generates the hierarchical links based on their physical card locations. This allows users to rearrange the Card Wall temporarily and easily restore the Card Wall arrangement. This method of organization could be coupled with the non-dimensional method which allows for an easier adaptation to the electronic Card Wall's functionality from it's traditional use.

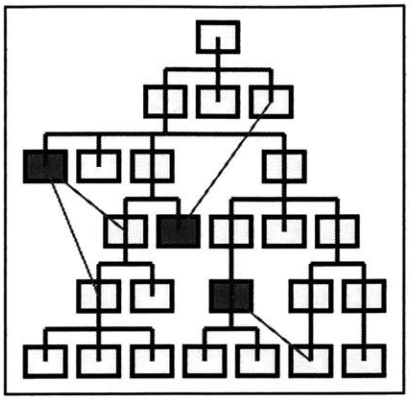


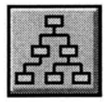
Outline Arrangement Figure 21

Outline (Hierarchy)

The outline mode is a feature found in most word processors and window managers. An initial outline arrangement is a list of top headings. The sub-headings of each heading can be viewed individually and in direct relation to its heading and other sub-headings on the same level. The hierarchical structure of an outline is visualized through the horizontal displacement of sub-headings. Figure 21 illustrates a possible Outline arrangement on an electronic Card Wall. The shaded cards represent headings. The cards that belong to a heading are lined up to the right of the heading. Heading cards have arrows attached that indicate whether or not a sub-heading is displayed. The arrows also serve as buttons for the activation and deactivation of sub-directories. This arrangement allows interactive searching of cards and the comparison of cards on the electronic Card Wall.

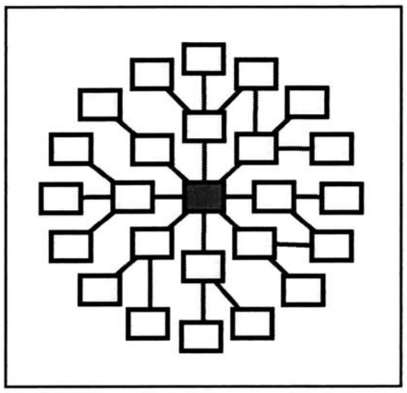
Unlike the Card Wall arrangement, the Outline arrangement allows for the visualization of both, hierarchically linked and networked linked cards. This is because network links can be made visible through replication. The infinite recursion problem mentioned previously is of no negative consequence to the Outline arrangement, since the user decides manually which and how many sub-levels he wants to display. Another advantage over the Card Wall is that it allows isolation during comparison. This is because the content of two sub-directories can be located next to each other and therefore directly compared (compare with the matrix arrangement).

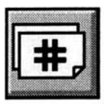

Hierarchy Arrangement Figure 22



Hierarchy (Hierarchy)

So far we have learned about two types of arrangements that allows for the visualization of hierarchical structures (Card Wall and Outline arrangement). The Hierarchy arrangement is the third arrangement of this type and probably the most traditional representation of a hierarchical structure. The Hierarchy arrangement draws the hierarchical tree from top to bottom with the application of various techniques for horizontal and vertical optimization. Similar to other hierarchical arrangements, this arrangement requires the user to select a top and to decide whether only hierarchical links must be considered. This arrangement proves its usefulness in the search for emerging hierarchical patterns on the Card Wall. Figure 22 illustrates an example of a hierarchical tree. The shaded cards indicate the end of a leaf. The non-orthogonal lines are left-over links that did not match the hierarchy.

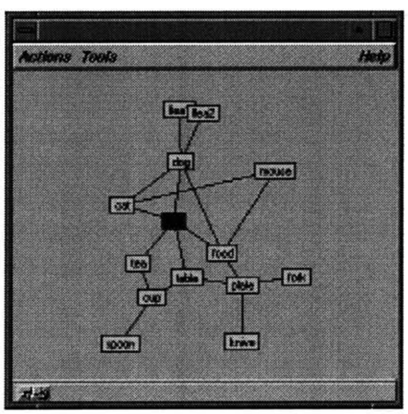

Center Arrangement Figure 23



Center (Hierarchy)

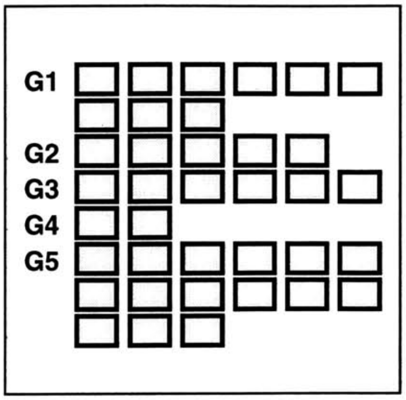
The Center Arrangement takes a pre-selected card and arranges immediately connected cards in a circle around it. A second circle gathers the cards that are connected to the cards of the first circle. This arrangement allows for an enhanced visualization of closely related cards. More than two card circles may be displayed at a time. This arrangement visualizes the hierarchical order of a few card levels with as few as possible intersecting links.

A different solution to this problem was realized computationally Yoshiaki Araki. His Java applet automatically assembles connected boxes around a pre-defined center-box. The boxes are in motion and keep moving until a solution is found with as few intersecting links as possible. The constantly changing pattern of boxes allows for the recognition of box combinations by chance.



Graph Layout by Yoshiaki Araki Figure 24

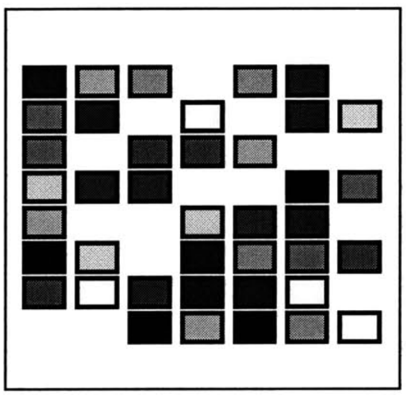
Both types of arrangements allow for the fast and interactive comparison of linked cards. While the previously explained arrangements reorganized the whole Card Wall, the Center arrangement only isolates a few cards. This visualization is probably more effective if displayed in a separate window in addition to the Card Wall.



Group Arrangement Figure 25

Groups (Group)

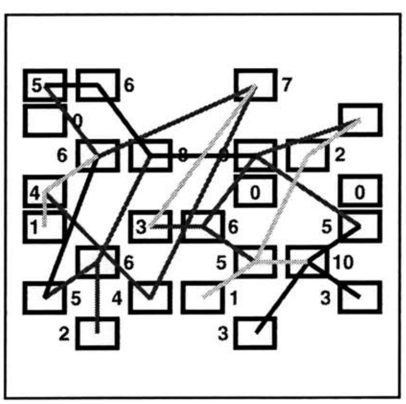
A group is a cross between a hierarchy and a layer. To a user, it is a matter of preference whether one works with hierarchies, groups, or layers. The simultaneous use of all these is confusing and in most cases not advisable. The Hierarchy Arrangement assigns cards to interconnected headings and sub-headings. Layers on the other hand, assign cards to individual headings. A layer is basically a one-level hierarchy. A layer can contain many cards. A card can only be assigned to one layer. Groups consist of cards that are attached to an individual heading card. Group cards can not be linked together and a card can be attached to more than one group card. Grouping is a very traditional and uncomplicated way of organizing data. A free exploration of ideas can proceed the grouping of cards. The grouping of cards can be a preparation for a Card Wall arrangement where the groups later become the headings or sub-headings. The Group Arrangement lists groups of cards in rows. The rows can be rearranged to allow for the direct comparison of cards in two different groups.



Priority Arrangement Figure 26

Priority (Network)

Often it is unclear whether or not a concept is important enough to be displayed on the Card Wall. Concepts that initially seem unimportant might become valuable at a later stage of the development process. Use of the traditional Card Wall forces an immediate decision about the value of an idea or concept. Concepts are either transformed into cards or left out. The electronic Card Wall allows the assignment of priorities to cards and links. A seemingly weak concept can be transformed into a card with a low priority which can later be modified. The electronic Card Wall encourages the user to transform as many concepts as possible into cards by delaying an immediate decision about the value of the card. Card priorities also indicate issues which need to be considered first. Priorities can be visualized in many ways. For example, we can group cards of equal priority or blend cards of low priority into the background. The possibility for automatically defined card and link priorities is discussed in relation to the Highway Arrangement.



Highway Arrangement Figure 27

Highway (Network)

The Highway Arrangement is based on previously discussed concepts concerning the Unified Card Wall and the Priority Arrangement. The Unified Card Wall introduced us to the basics of how to read links while the Priority Arrangement implied how to manipulate card and link values. The Highway Arrangement automatically generates and visualizes priorities based on the links assigned to cards. Figure 27 illustrates three different types on how card and link priorities are calculated.

The card priority is defined as the sum of all links attached to a card, in addition to the user defined card priority. The card on the bottom right for example, has four links attached (+4). Two of those links have no user assigned priority (+0), one link has a priority of three (+3), and one link a priority of two (+2). This assigns a priority of 7 to the card. A link priority is calculated by adding the user defined card priorities of both attached cards. To define both the card and link priorities, one first computes the card and then the link priorities according to the previously described method. The user defined values are always added to the total of the calculated value. This allows users to influence the automatically generated value or to assign a priority manually.

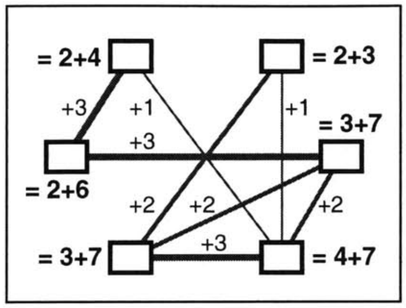


Figure 28 Automatic Defined Card Priorities

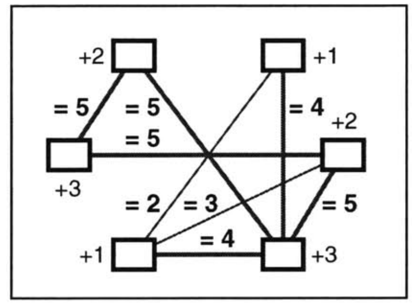


Figure 29 Automatic Defined Link Priorities

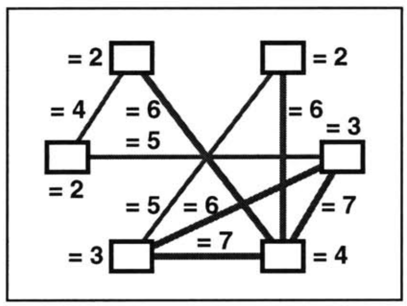


Figure 30 Automatic Defined Card and Link Priorities

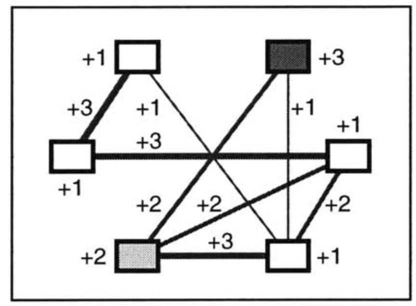
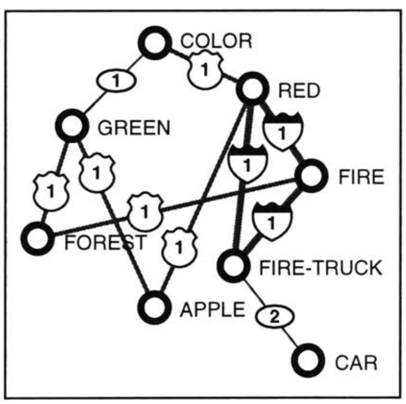


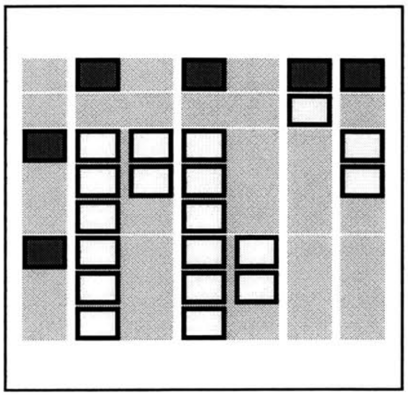
Figure 31 User Defined Priorities



Networked Information Figure 2

The Highway Arrangement visualizes link priorities by changing the thickness or the color of links. Figure 2 illustrates an earlier graphic. The line thickness of the network is calculated according to the previously explained rules. The street symbols (Interstate, Highway, Freeway, Road, Street, Way) represent a series of connected links of equivalent thickness (priority). If two link series of equivalent thickness exist or if a link series becomes separated the symbol shows a different number (corresponding to street maps). This visualization allows for the recognition of the most heavily linked cards and their interrelations. It encourages users to think of relations between ideas in a different way and to recognize idea groups of equivalent weight.

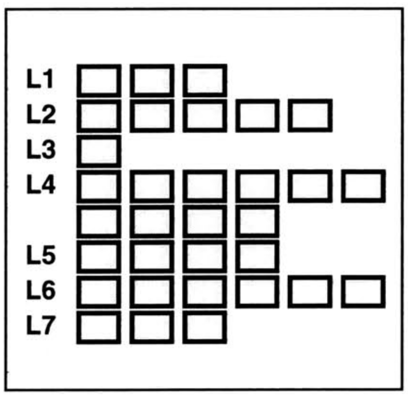
The mental process of relating information is called *brain mapping*. Let us consider a few possible relations between brain mapping and the previously described visualization technique. Some typical psychological tests include questions such as: "What comes to your mind when you hear the word 'fire'?". A possible answer might be the word "red". The psychologist would go ahead and ask: "What comes to your mind by the word red?" The answer might be "color". The psychologist would continue to ask this sort of questions and trying to find the words that are heavily related. We would call this process a search for Interstates that connect the most heavily used knowledge sections in somebody's mind. Since little is known about brain processes, we can just assume that we store knowledge in a similar fashion and that the visualization of information in a Highway Arrangement is helpful for to our understanding.



Matrix Arrangement Figure 32

Matrix (Network)

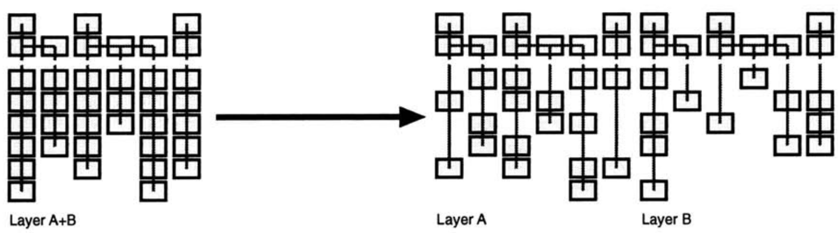
The Matrix arrangement is the Card Wall arrangement originally proposed by William Pena. It is similar to the Card Wall arrangement used by Henn Architects. With this arrangement, sub-headings are not placed below headings but along the y-axis of a predefined grid. This arrangement takes up somewhat greater physical space. It allows for the comparison of cards that belong to the same sub-heading but not to the same heading. Consider the following example: Both headings "USA" and "Europe" contain the sub-headings "Cars" and "Buildings". The matrix allows for the direct comparison between the cars and buildings developed in both countries. The disadvantage of the matrix is that it only allows for one level of sub-headings (two axes). Another important difference between the two arrangements is that the Card Wall arrangement is hierarchical and the Matrix relational. A spread-sheet for example, can not be expressed with a Card Wall. Likewise, an organizational tree structure cannot be expressed with a Matrix arrangement. It is therefore important that both arrangements coexist.



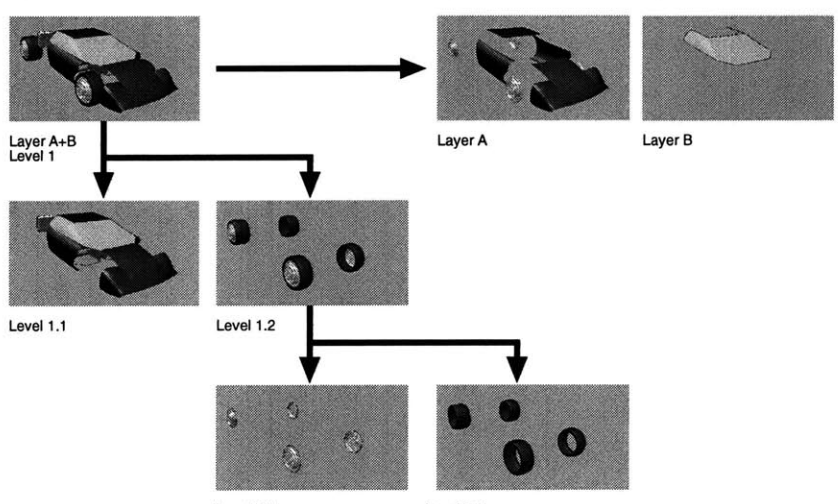
Layer Arrangement Figure 33

Layer (Network)

Layer arrangements are well known from CAE drawing programs. Cards can be assigned to different layers or switched between layers. Users can hide, freeze, or rearrange layers. *Hiding* a layer makes the cards on a specific layer invisible. *Freezing* a layer prevents from accidental changes. *Rearranging* layers gives the user control of what layers are drawn in front. This is a necessary option if cards will overlap. Later, we will discuss the layer management in relation to the user interface of the electronic Card Wall. Figure 33 illustrates the possibility to list layers with their content. Figure 34 presents an example of the activation and deactivation of layers.

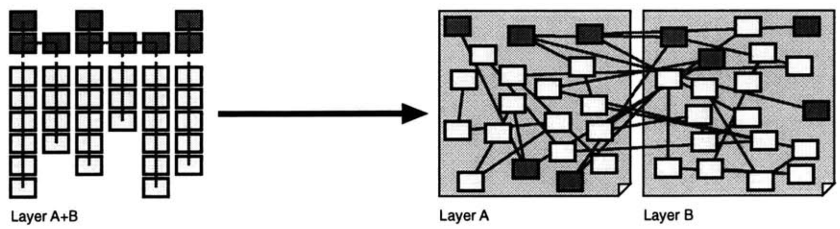


Layer Separation Figure 34



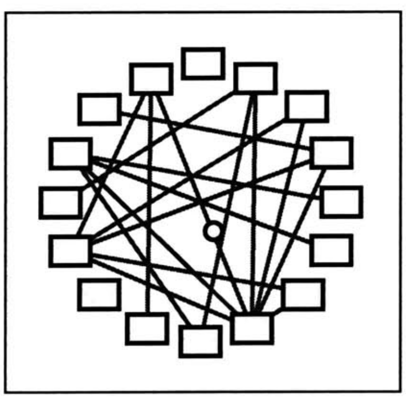
Combination Hierarchy/Layer Figure 35

Working with links and layers simultaneously involves several advantages. Figure 35 illustrates a few hierarchically linked car parts that are arranged on different layers. In some cases, the designer works only on one specific part of the car. This may be parts such as the wheels or the chassis. The different car parts are viewed and modified through the activation of a specific hierarchical level. The top level is viewed to check for possible conflicts between car parts. This is parallels the ability of the Outline arrangement to compare specific sub-parts of a structure. We learned about similar opportunities with cards in relation to the Outline arrangement. The additional use of layers allows for the grouping of cards that share certain properties. Figure 35 for example isolates all the metal and glass parts of the car.

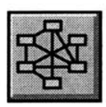


Hierarchy/Layer Separation Figure 36

Moving and linking cards on an electronic Card Wall is fairly easy and allows for various comparisons among groups of cards. However, as more cards and links are created, the arrangement becomes more complex. Layers allow for the separation of cards that are not of present concern. The cards do not have to be erased or moved but only assigned to an invisible layer. Cards on a common layer can also be linked together automatically.



Circle Arrangement Figure 37



Circle (Network)

Linking cards becomes a problem if many links exist or if the cards are packed closely together. Imagine three cards in a row and a link that connects the first to the last card. Since the link crosses the second card, it is not clear whether the first card is linked to the second card or to the third card. The user would have to move the second card to find out about the destination and origin of the link. A circular arrangement of cards is the only arrangement where links never cross other cards. The Circle arrangement is therefore useful for linking cards and viewing links among them. The Circle arrangement can also help to visualize cards that are linked more heavily. The disadvantage of the Circle arrangement is that it requires a lot of space. In addition, links between adjacent cards are sometimes difficult to recognize because they may be hidden by other cards. This problem is partially solved by maximizing the sum of the length of all links. This means placing linked cards in opposite positions on the card circle moving the *center of gravity* between all links as close as possible to the center of the card circle. The computer can be of support by either arranging the cards automatically or by indicating the center of gravity between links. Figure 37 illustrates a Circle arrangement with the center of gravity between links.

The Circle arrangement is also a helpful tool for the recognition of network patterns within a group of cards. Consider the following example concerning communication. A division of the MIT Sloan School of management under the supervision of *Tom Allen* developed a software program called *Netgraph*. Netgraph allows for the visualization of communication within companies in a two-dimensional matrix. The Netgraph illustration in figure 38.1 visualizes the interaction among 22 people of 3 teams. All 22 people are listed along the x and y axes of the matrix. The grid lines separate teams. The interaction between two users is visualized by a dot on the matrix. If communication occurs both ways (e.g. user A talks to user B and user B also talks to user A) two dots are drawn in mirror position of the diagonal from the top left to the bottom right (one dot on the intersection of row A and column B and one dot on the intersection of row B and column A). Therefore, if communication always occurs both ways, the pattern on one side of the diagonal mirrors the pattern on the other side. Since users don't talk to themselves, the diagonal itself contains no dots. The communication between teams may be examined accordingly. The total dots within a team box indicates the interaction within a team or between teams. Different colors are used to indicate supplemental information. The colors of the dots in figure 38.1 for example, might separate communication by gender. The Netgraph is a valuable visualization tool that allows analysis of communication between people and teams. In some cases however, alternative representations might be easier to understand. Figure 38.2 illustrates the communication between teams by using links of different thickness and color. The bright lines reference the bright dots in the Netgraph and suggest an equal spread of communication between the three teams. The dark lines indicate less communication between teams A and B. Figure 38.3 illustrates the communication between the members of team B in a Circle arrangement. This arrangement is easier to analyze but requires many different visualizations.

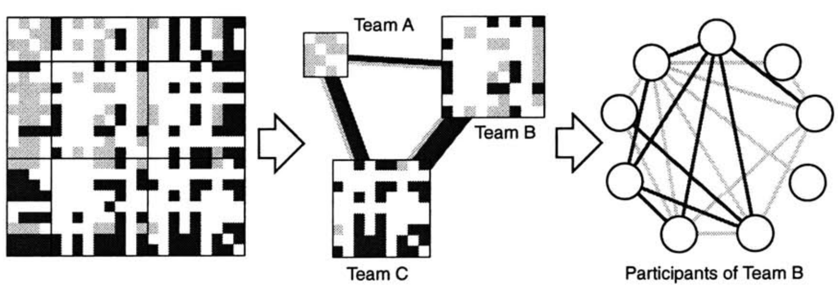
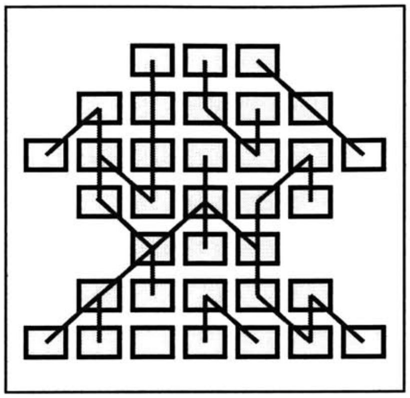


Figure 38
Alternative Netgraph Visualizations

Similar to the previously described Matrix arrangement, the Netgraph does not visualize relations among pieces of data using links. This makes the Circle arrangement an important addition to the electronic Card Wall. The Circle arrangement not only allows for the visualization of communication patterns but for the visualization of networked information in general. Quantitative, qualitative, and directional information can be expressed by changing the line thickness, the line color, or the arrowheads of links.



TimeLine Arrangement Figure 39

Time Line (Progress)

The Timeline arrangement visualizes the Card Wall progression over time. This function is not supposed to help people organize data, but to help them analyze why they did what they did. To understand some of the functionality and the underlying idea of this arrangement, we first discuss one more theory on brain mapping.

Initially, we analyzed various design *processes* and *methods*. This section will introduce two design *techniques*. The *Analysis* technique is more analytical and the *Synthesis* technique the more creative design approach. Both techniques have their advantages and limitations. Design is often a combination of both, analysis and synthesis.

Analysis

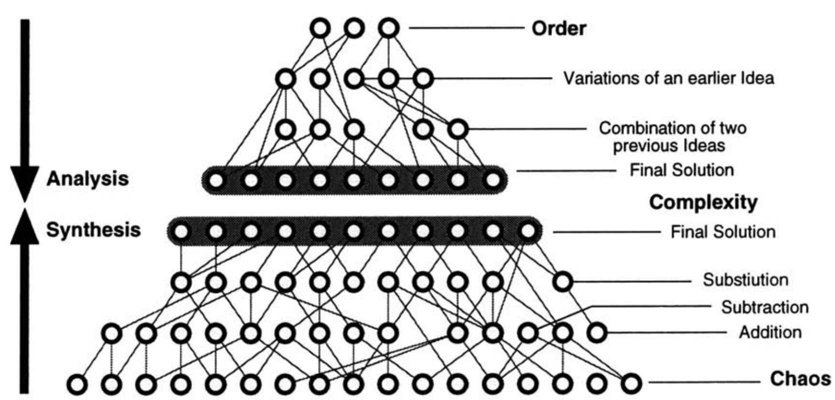
The analysis approach discovers possible solutions or partial solutions to improve something which already exists. The more possibilities discovered, (that can also be based on previously discovered possibilities) the more complex the solution becomes. The final result is a small but precise improvement over the existing solution. The analysis approach is a very common procedure for engineering tasks. A new car model is not developed by reinventing the car but by considering some refinements in aerodynamics. The more improvements which are considered simultaneously, the more complex the task becomes. Figure 40 illustrates this technique as the transition from *order to complexity*.

Bottom-Up	Top-Down
Creative	Rational
Generative	Diagnostic
Case-Based	Rule-Based
Random	Empirical
Artistic	Scientific
Inductive	Deductive
Linear	Generative

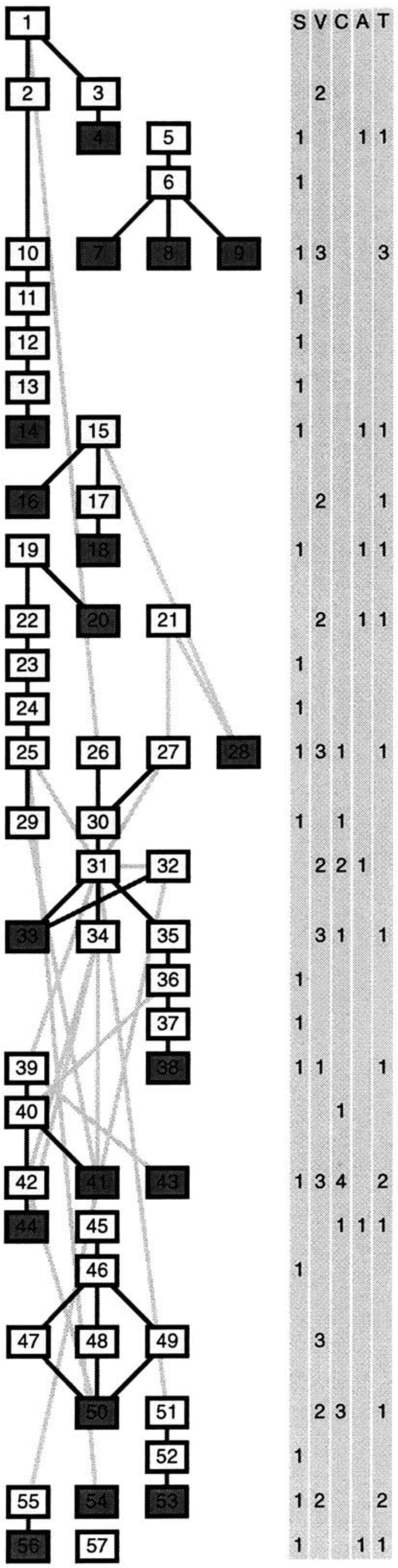
Methodologies Figure 41

Synthesis

Unlike the analysis technique, the synthesis technique starts up with the free exploration of possibilities. The range of possibilities is narrowed down through the analysis, combination, and improvement of previously discovered possibilities. This technique allows for more innovative solutions. The task remains partially solved if the range of possible solutions can not be narrowed down to an manageable level of complexity. This approach is more familiar to designers and inventors that search for alternative, revolutionary ideas which vary greatly from the status quo. Figure 40 illustrates this technique as the transition from *chaos to complexity*.



Problem Solving Techniques Figure 40



Progress Visualization Figure 42

The dots in figure 40 symbolize possibilities, solutions, ideas, or cards. Each row of dots represents an evolutionary step in the design process. Dots transform in five different ways: 1) Two or more dots that originate from one dot are called variations (V). A variation, for example, can particularize a general idea. 2) A combination (C) is a unification of two or more dots. Combinations can be conclusions from previous ideas. 3) A substitution (S) is merely a refinement of an earlier idea. 4) Ideas that are not further considered are called subtractions (T). 5) Finally, new ideas, that do not conclude from previous ideas, are called additions (A).

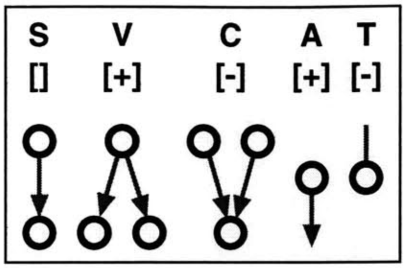
The analysis and synthesis techniques represent two different extremes. Figure 41 lists a few expressions that define similar distinctions. The successful completion of a task often requires the combination of various problem solving techniques. One possible way of combining different problem solving techniques is by bringing people of different backgrounds together. Their combined effort allows for the shared exploration of ideas in various ways. The Card Wall allows for various techniques to be used simultaneously by many people.

Progress Visualization

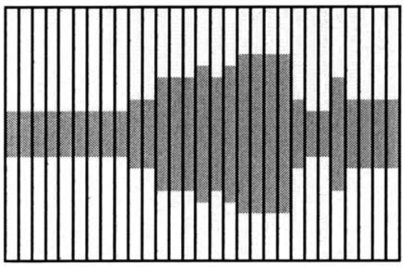
The progression of the Card Wall is visualized through the analysis of the card chronology and the classification of links (Substitution, Variation, Combination, Addition, Subtraction). Consequently, this visualization assumes time-stamps on cards. Time-stamps can be assigned automatically and require no user input . However, for an accurate visualization of progression, users need to link cards together as frequently as possible.

Figure 42 contains cards created by an student during an actual brainstorming session. The student envisioned the content and structure of a written paper by randomly generating cards containing some of his ideas. He was supposed to compare every new card with the previously created cards and to check for the possible relations. The student visualized the relations among cards with hand drawn lines. All cards were randomly placed on the board and numbered chronologically. The diagram in the figure was created after the brainstorming session. The numbers indicate the chronological order of the cards. Cards are organized in order from the top to the bottom. To conserve space, more than one card is placed in a row in the case that none of those cards are linked to each other. All links are restored as originally created. Black links connect consecutive cards while shaded cards indicate "subtractions".

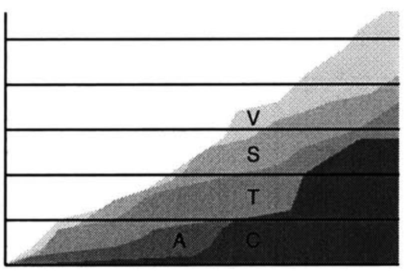
The five columns labeled SVCAT (Substitution, Variation, Combination, Addition, Subtraction) contain the total amount of cards of each type. For example, Card 2 and 3 are variations of card 1. This adds 2 *progression points* to the "variation" column for those two cards. The next row contains one card that adds one progression point to the "substitution" column, one card which adds to the "subtraction" column, and one card that adds one progression point to the "addition" column.



Progress Evolution Figure 43
S = Substitution; V = Variation
C = Combination; A = Addition
T = Subtraction



Progression Meter Figure 44

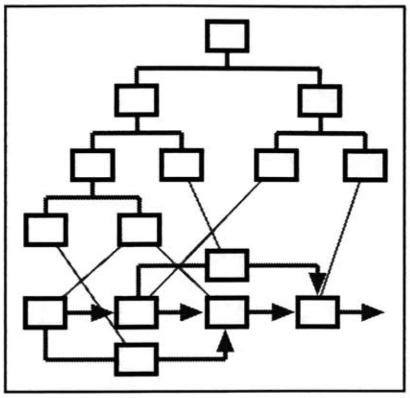


Progression Counter Figure 45

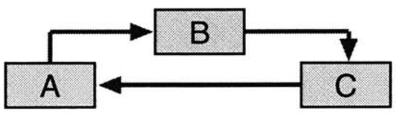
The *progression value* of each row is calculated as the sum of the progression points in column V and A minus the progression points in column C and T. Figure 43 visualizes the five types of progressions and their influence on the total progression value. The *progression meter* in figure 44 displays the total of all previously calculated progression values. The *progression counter* in figure 45 represents the total progression points of the individual columns.

If we analyze this progression of ideas in more detail, we find that the student initially had a very clear concept about the main thrust of his text. The first few cards developed hierarchically and have no cross links. This means that the student was unlikely to discover something new at this point, but expressed what he knew already. Since initial ideas advanced from card 1 and terminated with card 7,8,9, and 14 we assume that for some reason the student changed his mind about the main ideas of his text (In reality, the student later explained that the initial idea of the text became part of the introduction while later ideas formed the main body of his writing). Notice that Card 15 is an initiator of an idea that terminated immediately after. In this period, the progression meter does not indicate activity during the first part of the development and the "combination" graph in the progression counter (that represents conclusions from previous thoughts) shows no increase in value. After card 19, we find a more complex structure of links among cards which causes a n increase in the progression meter. Because the student has constantly linked more recent cards with earlier cards, the progression meter does not exceed certain limits. A constant increase on the progression meter would indicate that previous cards are not reconsidered properly or that the subject is constantly changing. A low value on the progression meter indicates that either the development process has come to an end (ideas are all properly linked together) or that new ideas have not been provoking further considerations.

It is unclear what changes on a progression meter indicate positive or negative development. We can assume that a repeated increase or decrease on the progression meter (within certain limits) indicates a "healthy" development. In it's purest form, an "analysis" approach (from order to complexity) is more likely to cause a permanent increase on the progression meter while a "synthesis" approach (from chaos to complexity) is more likely to yield a decreasing pattern. Ideally calculation of the progression meter is dynamic; that is, an previously drawn progression curve changes if cards get linked to previously created cards. This is very possible with a software implementation.



Dependency Arrangement Figure 46



C close to A and A far from C Figure 47

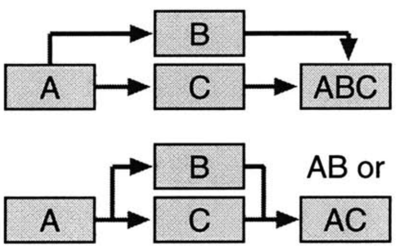
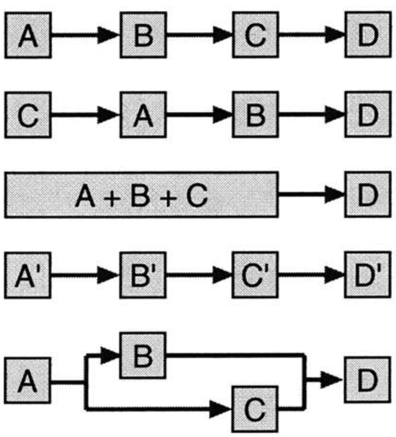


Figure 48
Multiple and Optional Dependencies



Process Transformation Figure 49

Dependency (Process)

Two objects (cards) that are connected by an arrow headed link express a dependency. Typical examples of diagrams that express dependencies are flow-charts, graphical representations of time schedules, or visualizations of business processes. *Time* is an important issue concerning dependencies. If for example an object A depends on an object B then object A must execute (or at least exist) before object B. *Location* is also of concern to dependencies. If object A depends on object B and object B depends on object C then object B must be located between object A and C. *Direction* affects dependencies as well. Figure 46 illustrates an object A that is accessible by object C. C, however is not accessible by object A. Finally, figure 47 illustrates the concept of *multiple and optional dependencies*. In a multiple dependency an object depends on many objects while in an optional dependency, an object depends only on one object. The first example requires for its completion B and C, while the second example requires either B or C for completion.

This chapter will explore possible applications of dependency links in relation to the electronic Card Wall. Since dependency links express time, location, and direction they might permit a more dynamic use of the Card Wall.

Process Transformation

The concept of process transformation is best explained with an example. Figure 49 illustrates four cards A, B, C, and D. B depends on A, C depends on B, and D depends on C. Suppose that the four cards make up four steps in a coffee making process. Step A may be the preparation of the coffee cup, B the grinding of the coffee beans, C the boiling of the water, and D the final combination of all components. The first example in figure 49 suggests the *rearrangement* of A, B, and C. This process proceeds by boiling the water first, then preparing the cup, grinding the coffee beans, and finally combining the components. Another possibility is the *combination* of A, B, and C by considering a fully automated coffee making machine that requires nothing else but pushing a button. An alternative, and more common solution is the *replacement* of some of the processes. An example may be replacing the grinding of the coffee beans with purchasing pre-ground coffee. The last example of process transformation illustrates *parallel processing*. The coffee making process is made parallel if one person boils the water while another person grinds the coffee beans.

The possibility to rearrange, combine, replace, and parallel processes makes the electronic Card Wall a *process modeler*. This functionality encourages the breakdown and analysis of existing process structures.

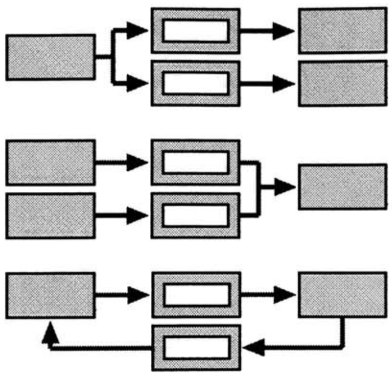
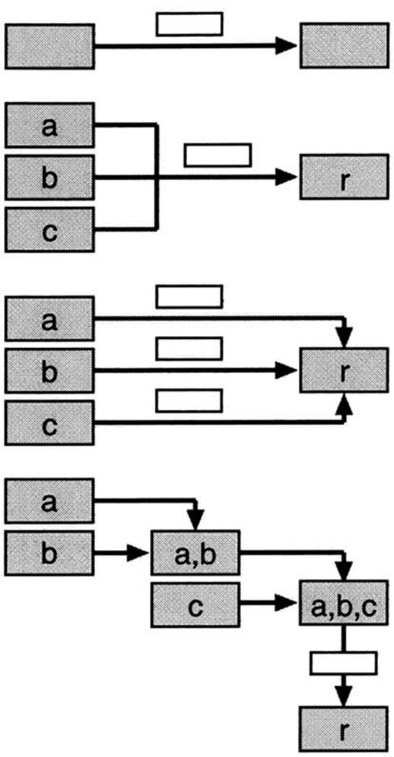
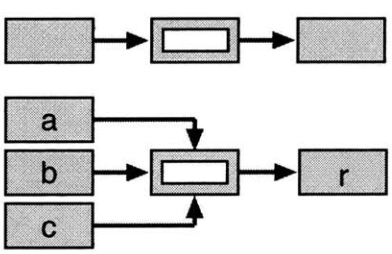


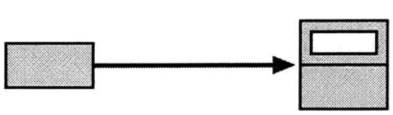
Figure 50



Function Links and Value Cards Figure 51



Function and Value Cards Figure 52



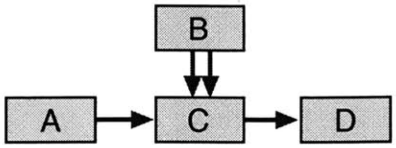
Function/Value Cards Figure 53

Process Visualization and Simulation

This section examines the possibility of expressing rules with dependency links. A *rule* is a function that transforms one value into another. Consider a dependency link of function f(x+2) that connects card A with card B. Card B would always contain the sum of the value assigned to card A plus the value 2. This makes the Card Wall a sort of calculator or spread sheet. The values of the cards are manipulated through process transformation or through the assignment of new values to some of the cards. Figure 50 illustrates three difficulties that offset this additional feature. 1) If the dependency links of a card point to two or more other cards and if each link contains a different rule, then all destination cards contain different values. 2) A card that links to another card that links back to the first card causes a loop with an infinite amount of possible iterations. 3) If two or more cards link to the same card, the rules are executed sequentially. This third example represents the most difficult case that needs to be studied further. As an example, suppose we have four cards that contain the variables of the expression r = f (a, b, c).

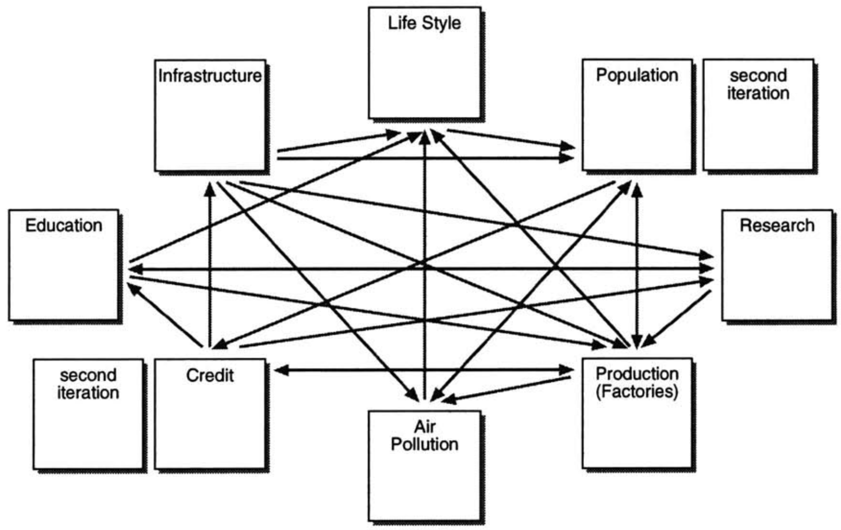
We will initially assume that cards contain only values and that links contain only functions. Since r is a combination of a, b, and c, we need to join three links and assign the function to the joint part of the links. Since the joint links are confusing and difficult to manipulate, we can alternatively consider separate links that all contain the same function information. Unfortunately this is not an optimized solution since information is unnecessarily duplicated. Another possibility is to obtain the result step by step. This would require the creation of two new solution cards. The first solution card contains the combination of a and b while the second solution contains the combination of the first solution card and c. Finally, the combination of the second solution card and the card c is expressed in r. This procedure does not elegantly solve the problem either, since unnecessary additional cards must be created.

Another possibility is to assign rules and values to links only. This causes less difficulties, since a, b, and c point directly to the card that contains the rule. The disadvantage is that we can not clearly separate between values and rules. Moreover, there is no obvious reason why the function card can not contain both, rules and values (as in a spread sheet for example). Only the assignment of values, rules, and results to cards limits the amount of necessary cards and allows for the easy compatibility with the rest of the Card Wall. However, using the card wall as a calculator still remains a difficult task. Consider four cards A, B, C, and D where the expression to be solved is A + (B - C) * D. Since we can not group cards between brackets on the Card Wall, the execution order of the expression must be defined through links. C must become the central element since it is first subtracted from B, then multiplied by D and finally added to A. To make sure that D is executed after B and before A, we either need to assign priorities to the links (this can be visualized by the link thickness) or by defining the order of execution within the card C.

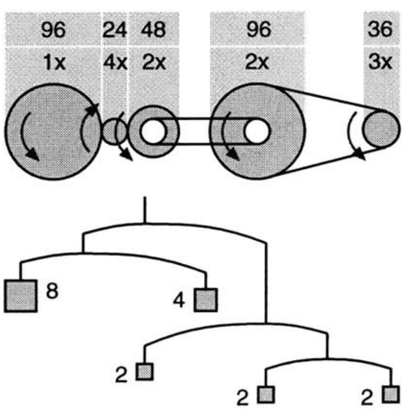


A + (B - C) * D Figure 54

A spread sheet can reference values from within an expression. This is not the philosophy of the *Card Wall Calculator* since the cards are supposed to hold the values and the links the references. It is therefore not advisable to compete with a spread sheet but to consider an alternative and more efficient use for the Card Wall Calculator. The strength of the Card Wall is the visualization of connections among objects. The Card Wall also allows easy changes on links and cards. Directional links visualize and support the understanding of complex systems. The dependency links on a Card Wall allow users to predict the outcome of a series of activities.



Economic System Figure 55



Visualization of Dependencies Figure 56

In a more general sense, the use of rules also applies to larger systems. Our economic system is a very complex assembly of dependency factors (rules). A few dependency factors are illustrated in figure 55. Improvements in infrastructure for example may cause an increase in quality of life, population, research opportunities, production, and air pollution. The money to improve infrastructure is obtained from the general population and the production industry. Figure 55 allows for the easy recognition of dependency factors. The significance of this influence is visualized through the differentiation in line thickness and color. The Card Wall Calculator could also test numeric examples over many iterations by computing the values assigned to cards. For example could we find out about the effect on the ecological system after an investment in education, infrastructure, or research. This is basically what a simulation like does. However, SimCity does not allow for alternative simulations, or for the visualization of dependency factors. The visualization of dependency factors allows users to *understand and predict the outcome* of a series of activities before a simulation is performed. Figure 56 illustrates two visualizations of a series of dependency factors. The first example uses gears to visualize the relations between five different values. All gears are interconnected and turn in different directions and with different speeds. The bigger gears at a different angular speed than the smaller ones. Two of the gears that differ in size turn with the same speed while the two other gears that have the same size do not turn with the same speed. The second example uses gravity as a tool for the visualization of dependency factors. All weights raise if the weight on the left increases. The weights on the left remain almost uninfluenced if the weight on the far right doubles.

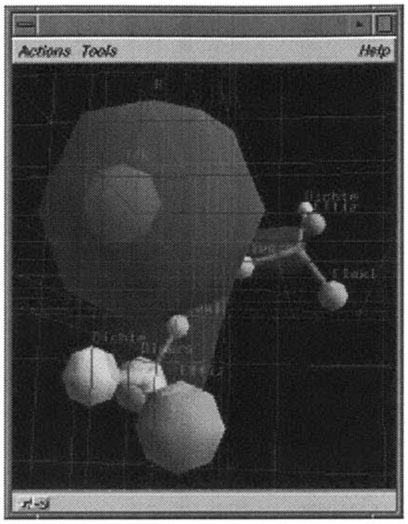
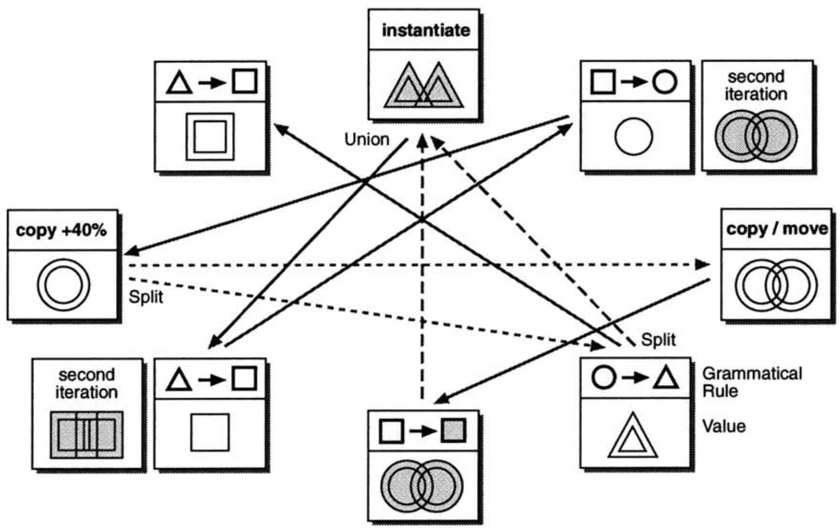


Figure 57
Planet Model by Wolmer Facchin

A more sophisticated model for the visualization of dependency factors was introduced by *Wolmer Facchin* (Research Associate at the ETH Zurich). The *planet model* in figure 57 visualizes the influence of variables in relation to other variables as planets in space. The planet's distance from the center of the model indicate the influence of a particular variable. Each of the planet's diameter corresponds to its weight. The system accomplishes the gravitational balance by turning around its center. The planet model may be viewed and manipulated in three-dimensional space. This allows interactive testing of the dominion of variables and dependency factors.

The understanding of the functionality of a simulation as well as the understanding of the interrelation among corresponding equations is achieved through the graphical visualization of values and dependency factors. The previous examples has illustrated three different ways of how to represent and manipulate variables and dependency factors. The following paragraph examines the possibility of conducting simulations by not only manipulating values and dependency factors, but the visualization itself.



Shape Transformation Figure 58

The Card Wall is known as a construct of cards and links. The cards can contain values and rules. The links can define dependencies between cards. The association of values, rules, and links on the Card Wall concludes in one or many possible solutions. Figure 58 illustrates cards that contain *shapes* and *shape typology rules*. The card on the bottom left instantiates a triangle with a square. The present value of this card is also a square. The card points to another card that instantiates the square with a circle. The next referenced card executes two rules. The first rule copies the circle and the second rule increases the size of the circle by 40%. This results in two circles of different size. This value is submitted to two cards simultaneously (*split*). The first card copies and displaces the circle pair and forwards the result to another card that creates the new shape. The second card instantiates the circle with a triangle. Both results are *unionized* by the top center card. This card instantiates the first value obtained with the second value. The two shaded circle pairs become two shaded triangle pairs. The following link points back to the first card and consequently starts the second iteration.

The shape changes with every iteration and often increases in complexity. Alternative results are generated by changing the values, rules, links, or amount of iterations. For efficiency reasons, a computational solution would only display the final shape in a separate window. (053) 058

An advanced example of computational form generation comes from *shape grammar theory* by William Mitchell and George Stiny. In this thesis, architectural shapes are generated by *replacing*, *adding*, or *subtracting* previous shapes. Since every shape can create many new shapes, the form generation process is hierarchical. The amount of possible solutions can grow infinite if intermediate results are not continuously deleted. The disadvantage of subtracting intermediate results is that valuable information may become lost. An alternative possibility is to narrow down the range of possible solutions with the *unification* of results. From *genetic programming algorithms* we know about this unification of two or more genetic codes samples. Shape specifications and grammatical rules can be combined in a similar fashion. This makes the Card Wall a tool for architectural form generation. The final result is manipulated by changing the content of the cards or the links between cards. The unification process is either user controlled (the user decides out of a given set of results which ones to combine) or computer controlled (the program decides randomly or based on pre-defined rules which intermediate results to combine). Possibilities for the genetic unification of architectural objects is illustrated in one of my earlier papers *on self-generating architecture*.

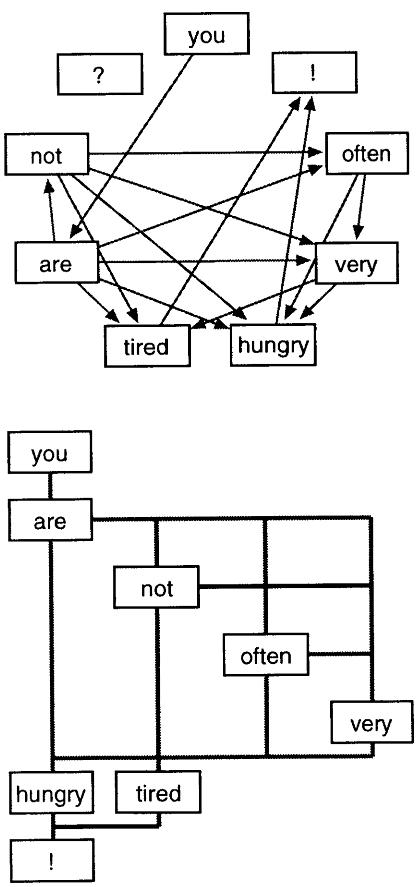
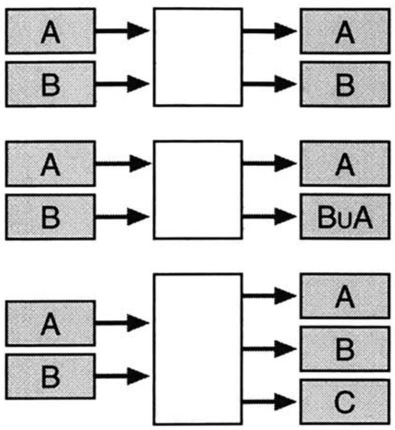


Figure 59
Visualization of possible word combinations

The difficulties with the free combination of vocabulary elements is the compatibility between rules and vocabulary elements. Signals which indicate impossible combinations must be attached to elements and rules. Figure 59 illustrates nine word cards that can be combined to form a sentence. The amount of possible sentences decreases every time a word is added to the sentence. Approximately half of the sentences that can be constructed from this set of words gets lost if the word "you" is placed at the beginning of the sentence. Only "are" can follow "you" if one of the rules requires a predicate after a subject. A computational solution could provide the user with a visual representation of all possible solutions (as illustrated in figure 59) or a list of all possible sentences. This would allow for the selection of the five possible options "not", "often", "very", "hungry", and "tired". However, the analysis of possible options in a more complex example could become a difficult task. In most cases it is more efficient and inspiring to link words randomly and observe the results. Too many results require the user to link more cards. Unsatisfactory results requires the user to change the existing link structure. The computer can also generate a customized result through the unification of user selected solutions or visualize the link structure that belongs to a user selected solution to allow for manual changes.



Agent Interaction Figure 60

Process Interaction

The previous examples have shown the various abilities of dependency links to understand, rearrange, and organize related objects. Such systems are helpful once we manage to create all of these dependency links. In some cases it is convenient to draw some conclusions before the creation of dependency links. Cards that have no links attached but contain information such as values, rules, or results are valuable to store data. Since scattered knowledge is of little use, the card information needs to be related in some way. We already explored various manual ways of linking cards. The following paragraph considers a computational solution that supports the linking of cards. The idea is to make cards interact and exchange knowledge. This concept is known from research in artificial intelligence on *intelligent autonomous agents*. Significant work on intelligent autonomous agents was done by Patty Maes. The basic idea is to make agents (cards) compare their knowledge pieces with the knowledge pieces of other agents and relate those knowledge pieces if appropriate. Figure 60 illustrates there possible outcomes from an interaction between two agents: 1) Two agents meet but do not exchange knowledge. 2) One or two agents accumulate knowledge during an interaction. 3) Two agents decide to create a new agent that contains the knowledge of both agents.

The following example is another expedition into "shape typology" and represents cards with LEGO bricks. Many different types of bricks are available and the rules on how they can be attached are known. Each brick is considered an agent that knows about the rules which dictate how to attach itself to another brick. Bricks meet by chance. If for example a blue brick meets a yellow brick, the blue brick can attach a copy of the yellow brick to itself. The yellow brick on the other hand might just continue its journey with the additional knowledge in mind that there exists a combination of a blue and a yellow brick. If this yellow brick later meets a red brick that wants to be attached to a blue brick, it may forward the information about the blue-yellow brick to the red brick. Both bricks might also decide to create a new brick that consists of a yellow and a red brick. The continuation of the interaction among the bricks leads to the generation of many shapes which later might join into one shape. The usefulness of the generated form depends on the quality of the previously defined rules.

The previous example does not necessarily illustrate the potential of this method but provides a basic understanding of its functionality. The intention is to allow cards to compare themselves with other cards, goals, or predefined concepts. Cards created in past sessions can forward their knowledge to recently created cards. Card patterns of different sessions can be compared and possible relations visualized. The underlying intent of such a system is the creation of an *Intelligent Card Wall* with the ability to learn.

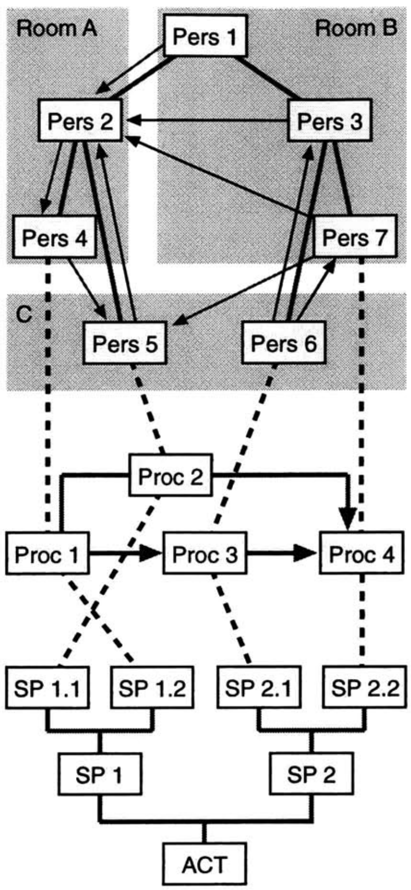


Figure 61
Combined visualization of processes, communication patterns, and hierarchies

Process Organization

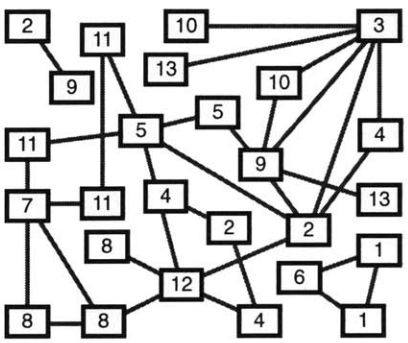
The ability to organize, save, and compare industrial and workflow processes is an important challenge for modern business. In this interest, there have been attempts among academicians to create tools which focus on the ease of visualization and storage of different processes. Ease of visualization allows for processes to be compared, while storage allows processes to be saved in an effective and meaningful way. An example of this is the *process handbook* by *Tom Malone,* which describes a system that divides a process into sub-processes. The processes are structured hierarchically under pre-defined headings. This allows people to compare, replace, and reassemble processes. The hierarchy of the process handbook is structured by having a parent process called an "act", which covers all possible processes. The next lower level includes sub-headings such as "create", "modify", and "use". "Create" for example is further sub-divided into "what" and "how". The pre-defined structure of the process handbook makes its incorporation into the card wall valuable. The card wall can be a useful addition to the electronic Card Wall because the Card Wall already allows for the arrangement of cards (or sub-processes) in various hierarchical forms.

For the Card Wall, it is important to be able to recognize mixed types of information organization. Processes within the Card Wall need to be recognized even though some cards within the process may be part of a different type of a structure. Figure 61 illustrates the combined visualization of processes, communication patterns, and hierarchies. This example shows person 1 in charge of person 2 and 3. Person 2 is in charge of person 4 and 5. Person 3 is in charge of person 6 an 7. The communication between people is visualized with thin arrows and differs from the hierarchical order. Person 4 for example hands off information to person 5 which follows from person 5 to person 2. The physical location of the people is expressed by the gray shaded areas. Person 2 and 4 for example work in room A. The process visualized below consists of four sub-processes, two of which are executed in parallel. The dotted lines reference the person in charge of the process. Person 4 controls process 1. The bottom of the picture illustrates the hierarchical order of sub-processes according to the process-handbook. This example indicates the complexity that arises from the combined visualization of many correlated systems. The Dependency Arrangement could allow for the automatic rearrangement of processes, for the visualization of sub-processes within the process handbook, for the indication of conflicting sub-processes, or for the visualization of the "critical path". Various computer applications that allow for similar activities already contain many of these capabilities. (056) 061

Working with dependencies among cards opens up many opportunities. Processes lead to the discovery of a variety of possible applications and an understanding of the complexity involved. Working with dependencies is obviously an interesting addition to the electronic Card Wall. Because of its complexity, this research project is of similar magnitude to the design of the electronic Card Wall itself.

GENERATED GROUPS

Some of the substantial features of the Card Wall are that many people can work together on one task, participate locally or remotely in a discussion, or discuss different issues in parallel. Working in parallel is of importance for several reasons. 1) People have different interests and strengths. 2) Group discussions usually contain issues that are not of interest to all participants. 3) Valuable ideas get lost while participants wait for a topic to be discussed. A large task is usually broken down into smaller parts and analyzed individually by different groups of people. The conclusions from those individual discussions are later compared among different groups. This chapter explores possibilities to automatically determine how people of common interests and abilities group themselves through the analysis of past Card Wall sessions.



Randomly Dispersed Cards Figure 62

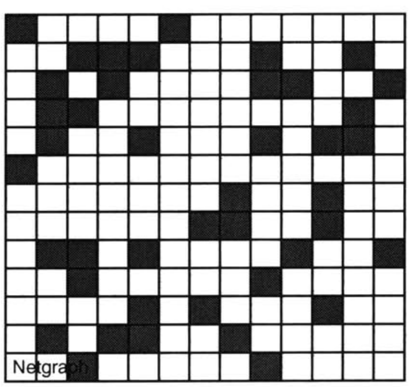
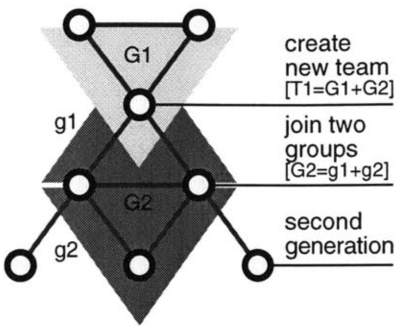


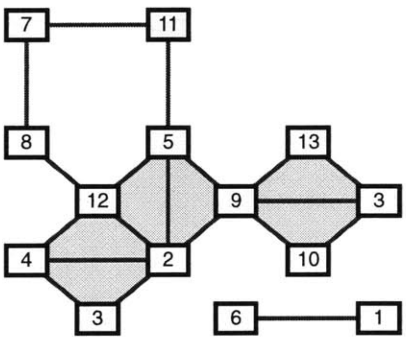
Figure 63
Netgraph Visualization of Card Links

The previously discussed Netgraph application allows for the visualization of communication patterns among individuals and groups. The Netgraph can also be used visualize the contents of a Card Wall. Since every card contains a thought by a person plus the name of the person, the links among cards connect the thoughts of different people. From these connections it is possible to visualize some sort of interaction. A Netgraph created from a Card Wall lists cards along the x- and y-axis and represents links between cards with dots on the grid. Alternatively we can list all card owners along the x- and y-axis. This reduces the amount of rows and columns, since most participants create more than one card. It also shows relations among group participants rather than card content. Figure 62 illustrates a few randomly linked cards. The numbers on the cards reference the card owners (person who created the card). The two cards on the top left for example are created by person 2 and 9. The Netgraph in figure 63 visualizes the link between the cards of person 2 and 9 with one dot in column 2 row 9 and one dot in column 9, row 2. The other dots are created accordingly. Each dot references two people who's cards are related in some way. The Netgraph also provides quantitative information about the linked cards rather than the total number of cards created by a user. This highlights users that have added valuable information to the Card Wall (assuming that a valuable card is a heavily linked card). The Netgraph in figure 63 does not take time into consideration. This is why the diagonal from top left to bottom right mirrors the Netgraph output. One could assume that the second card of a linked card pair came only into existence because of the presence of the first card. This would make the first card the more important one and represent the link between both cards with only one dot. However, time is probably not a helpful distinction for the judgment of the importance of ideas and is at this point will be of no consideration.

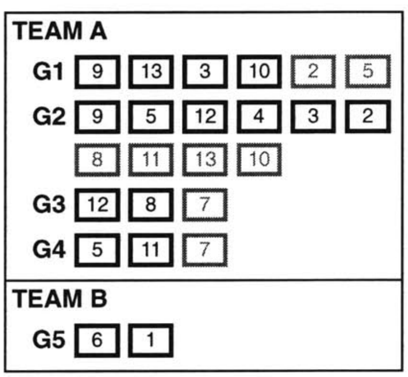
The previous paragraph explained Netgraph as a tool to recognize pairs of people who's cards are linked together. It is also of interest to determine groups of people who have cards which are linked together. These people share some understanding about a certain subject and should be identified to automatically propose a possible group arrangement. There are many potential ways to find out membership of groups based on connections. One way is that groups can be approximated under the application of the following few rules (compare figure 64): 1) A potential member can join an existing group if he created a card that



Group Generation Rules Figure 64



Group Recognition Figure 65



Group List Figure 66

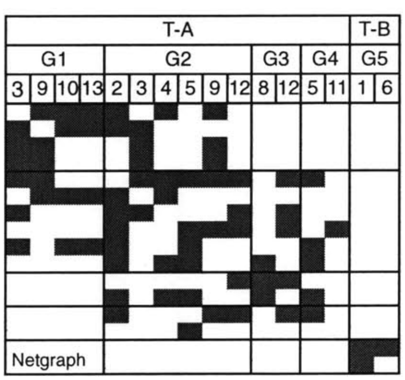


Figure 67
Netgraph Visualization of Groups and Links

was linked to at least two cards created by two different members of this group. 2) Two groups are combined together if rule 1 applies to two members of two different groups. 3) Two groups create a top group (for example a team) if they share only one member. 4) A potential member can also join a group if only one of his cards links with a card of a group member and if not more than one of his cards links to other places. Figure 65 illustrates the groups that could be generated from the card information in figure 64. A total of five groups are identified, two of which belong to team A. Three people are indirectly connected to team A and have a choice to join one of the groups in this team. Because two individuals are entirely disconnected from the other people, they not only create their own group but also make up their own team.

Figure 66 illustrates the previously created groups and teams in an ordered list. It shows that a person can be a member of more than one group. The cards with gray outlines represent people who are not considered to be full-members of a group. Figure 67 illustrates the links between the cards of group members in a Netgraph. This Netgraph lists groups rather than individuals along the x- and y-axis. The optimization of the previously mentioned group generation rules will force a concentration of dots along the diagonal of the Netgraph. This visualization can also be used as an observation tool during a Card Wall session. If dots get dispersed, the distribution of people within groups must be reassembled. One could imagine locating the members of a group next to each other on a table with the members of different teams on separate tables. Information exchange among teams is encouraged through the shared Card Wall content. Netgraph information might enforce direct communication among automatically assigned members of groups. In the future, the automatic group generator would not only generate groups from information obtained on the Card Wall but also from HTML submissions, e-mails, and phone calls. Considering the increasing opportunities for virtual collaboration on the Internet, the automatic group generator might support the visualization and formation of web-based communities. However, the primary use of the *automatic group generator* is to find out about people of common interest and to have them work on the same task when the Card Wall session is over.

THE ELECTRONIC CARD WALL

The previous study discussed many possible Card Wall functions and arrangements. This chapter is concerned with the development of the electronic Card Wall itself. Initially, we will analyze an existing program which provides some insight into the basic functionality of the electronic Card Wall. Next, we will do an investigation of needed attributes which explains some of the necessities as it concerns the program structure. The final illustration on user interfaces will explore possibilities that allow for easy handling and understanding of the proposed Card Wall functionality.

Basic Functionality

This paper is not concerned about the creation of cards. We will assume that cards are hand-drawn, digitized, and imported to the electronic Card Wall program. The cards illustrated in figure 68 were digitized with a digital camera and resized to approximately 10% of their original resolution. This allows for the simultaneous visualization of many cards on a computer screen. The full card resolution is used for print-outs only. The arrangement and linking of the cards in figure 68 was done in *Inspiration*, a program that provides almost all the basic functionality necessary for the electronic Card Wall program. 1) Cards can easily be imported to the program and placed on a customizable grid with snap mechanism. 2) Cards can be linked together by dragging a rubber band line with the mouse from one card to another. The rubber line becomes attached to the card and moves with the card. 3) Text excerpts can be attached to cards and links. 4) Cards can either be erased or crossed out. 5) An outline view can automatically be generated. 6) Arrangements can be exported to other programs. This small set of functions allows already for various improvements over the manual Card Wall.

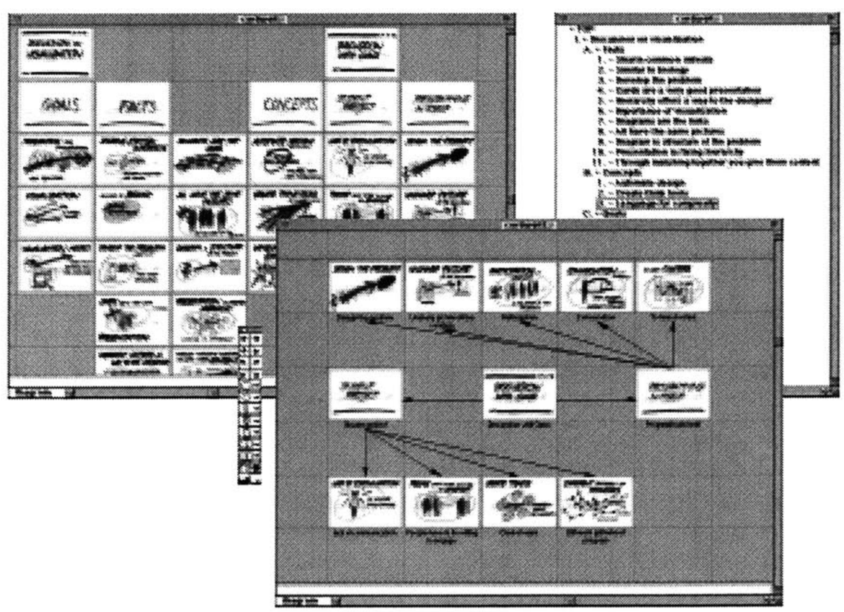


Figure 68
Card Arrangement in "Inspiration"

Attributes

Attributes are pieces of information that describe the properties of cards and links. The table in figure 69 lists in the first column all the attributes necessary for an electronic version of the Card Wall. The second column lists three different levels of computer support for the initial assignment of attributes. *Partial support* requires the user in some cases to specify an attribute. *Automatic support* makes initializes attribute information automatically (such as the time the card was created). *Default* values for attributes are provided in some cases and only changed by the user if necessary. The columns to the right of the support levels list the four different types of links and the two different types of cards which have been defined earlier in this paper. A more detailed explanation of the different attributes is given below.

ATTRIBUTES	SUPPORT			LINKS				CARDS	
	Partial	Automatic	Default	Network	Hierarchy	Process	Group	Heading	Standart
Name	■		■	■	■	■	■	■	■
Date		■		■	■	■	■	■	■
Time		■		■	■	■	■	■	■
Keyword								■	■
Picture	■								■
Text				■	■	■	■		■
Priority	■		■	■	■	■	■		■
Color	■		■	■	■	■	■	■	
Thickness	■		■	■	■	■	■		
Layer	■							■	■
Value									■
Formula					■	■			■
Reference								■	■
XY - Coordinates		■						■	■
List of Links		■						■	■
Link Destination		■		■	■	■	■		
Types			■	■	■	■	■	■	■
Arrow		■				■			

Card and Link Attributes Figure 69

Name, Date, Time

This information is important to the Time Line arrangement as well as to the automatic group generator. Only the names of remote participants can not automatically be determined by the system. Date and time are always attached automatically upon the creation of a new card. Both, cards and links contain name, date, and time information.

Keyword, Picture, Text

The visual card information consists of keyword, picture, and text. The keyword acts as a unique identifier for each card, which can be used for searches. Users always refer to a card by its keyword because the keyword is usually an indication of the card's content. The Picture is an iconic representation of the keyword that allows for easy recognition and memorization. Pictures are selected from a picture library or may be digitized and loaded. The text is optional and provides additional information about the card.

Priority, Color, Thickness

These three attributes are used as an indicator of importance. Color and line thickness are assigned to links. Priorities are assigned to both cards and links. The automatic assignment of priorities is discussed in relation to the Highway and Dependency arrangement.

Layer

Only cards may be assigned to layers. Links are part of the card information and thus are displayed on the same layer as its card. If a link connects two cards on different layers and if one of the layers is turned off, the link is not displayed. The detailed use of layers is discussed in reference to the Layer arrangement.

Value, Formula

The section on Dependency arrangements discussed the use of values and formulas with cards. Value and formula information on cards allow for the execution of calculations and simulations.

Reference

A reference connects a card to some external card information such as a text document, a construction detail, or a WWW site. A double-click on a card opens a document in its application. This option further expands the functionality of the Card Wall and makes it a tool for the *organization of shared documents in a common space*. The visualization of relations among shared documents allows for advancements in remote collaboration.

Coordinates, Link List

This information is only of computational concern. Every card needs to know about its position on the Card Wall. This allows users to save and undo card positions. Link information is always contained within cards. This allows for the exchange of card and link information between Card Wall's. If cards are moved to another Card Wall, the link information can be restored if cards having equivalent keywords exist on both Card Walls.

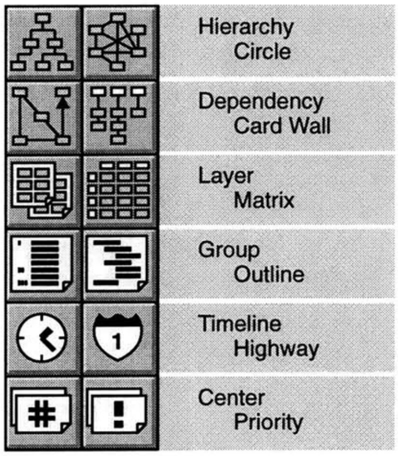
Type, Arrow

We previously examined different types of links and cards. The program requires the specification of card and link types to allow for advanced arrangement options. The standard card and the relational link are the most often used types and therefore set as a default. Arrows are attached automatically in the direction of the way the link was drawn if the dependency link type is selected.

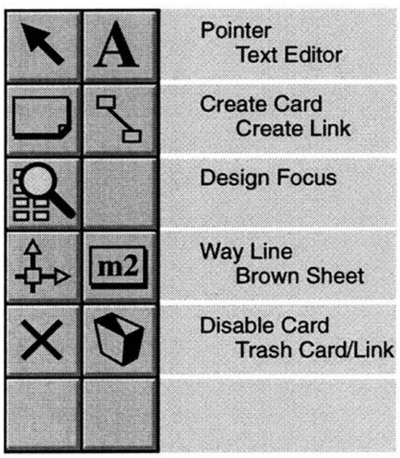
User Interface

The many attributes of the proposed Card Wall imply that the user has to deal with too many settings. Using the Call Wall may become difficult and confusing for users because of the vast number of features and their abstract nature. The efficient use of the electronic Card Wall is only guaranteed if a minimum of user specifications are required. In addition to being simple, the design of the user interface must be clear and well organized. For example, the basic use of the electronic Card Wall only requires the definition of a keyword for the creation of a new card. The creation of a new link can be left to the default values entirely. Additional specification of attributes increase the functionality of the program and can be done at will. The user will only take advantage of the full functionality of the program if the design of the user interface allows for an easy manipulation of attributes and cards.

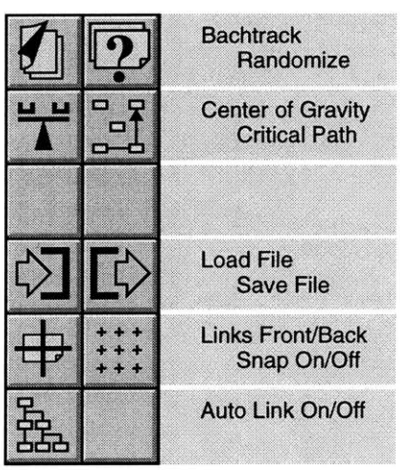
The initial design of the user interface do not include pull-down and pop-up menus. The entire functionality of the program is controlled through tool-bars (a tool-bar is an assembly of buttons and text boxes in a separate window). Similar functions are grouped together in separate tool-bars. The following paragraphs describe some of the proposed tool-bars.



Arrangement Menu Figure 70



Tool Menu Figure 71



Function Menu Figure 72

Arrangements

The first tool-bar includes the arrangement buttons. Each button rearranges the entire Card Wall according to the type of arrangement. The user is unlikely to loose track of certain card patterns since there are only twelve arrangements available. The user is likely to work with only one arrangement and use the others briefly to search for relationships among cards. All arrangements that require the selection of a card assume the last card moved or linked is the one that was selected. For example would one first select the top card of a hierarchical structure and then press the Hierarchy button. A later version of the program might also allow for the customization of Card Wall arrangements.

Tools

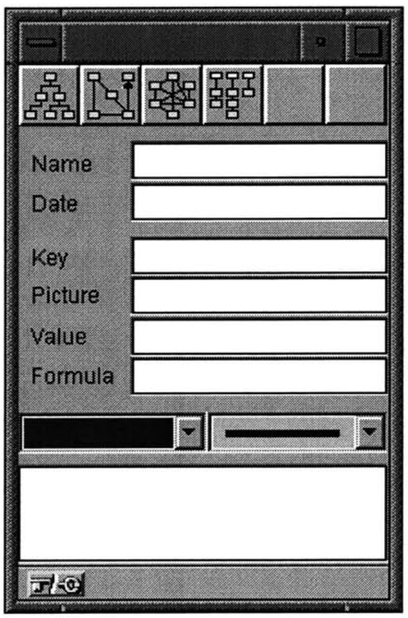
Tools allow for Card Wall manipulations. Because tool buttons are exclusive switches, only one tool can be active at a time. The "Pointer" button allows for *general manipulation* such as moving and re-seizing cards. The two switches below the pointer *add cards and links* to the Card Wall. The "Design Focus" button *zooms between levels of references*. We learned earlier about the possibility to reference files with cards. The "Design Focus" button allows users to view referenced information. It might launch a word processor, open a text file or link to another Card Wall. Card Walls can inherit other Card Walls as well. If for example a suggestion by one of the participants raises a profound discussion, the group members might decide to keep this piece of the discussion on a separate Card Wall. This new Card Wall is referenced by the card that initiated the discussion.

Dimensional information is attached to cards with the "Brown Sheet" button. This allows for the interactive evaluation of room sizes. Links created with the "Brown Sheet" button automatically *label links as room-connection links* that can be transformed into Way Lines. The last two buttons are used to *disable or erase cards and links*. Disabling a card means to mark it inactive. A disabled card becomes grayed out with a cross through it. Erasing cards and links moves them to an invisible layer. This layer can be viewed together with other layers. Erased cards are visualized in a different color and can be restored at anytime.

Functions

Buttons that cause an immediate action are called functions. "Back-track" for example allows a user *to undo* an action. "Randomize" *places cards in random positions* on the wall. This allows for a free, unorganized exploration of possible card relations. Another button allows for the visualization of the *center of gravity* among cards with assigned values. If for example, card A causes a change in card B and B causes a change in card A, then the center of gravity is located between card A and B. The center of gravity moves close to card A if A is of more influence to B than B to A. The *critical path* can be visualized with respect to time, if this information is assigned to cards (the critical path is known from PERT charts). Other buttons allow users to *save and load Card Wall arrangements* or to *hide links behind cards*. Viewing links on top of cards is important if cards are close together. Links are sent to the back if they are not of present concern or if they obscure card content. The following button turns the *snap mechanism* on and off. The snap mechanism allows for the precise arrangement of cards on a user defined grid. The last button controls the *automatic link generator* that was discussed in relation to the Card Wall arrangement.

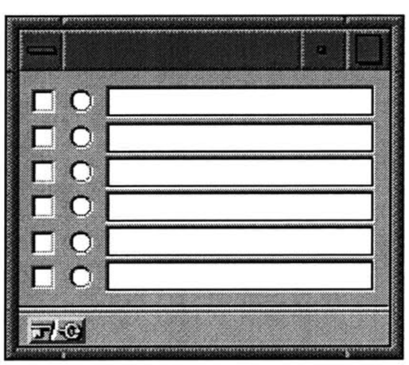
Attributes



Attribute Menu Figure 73

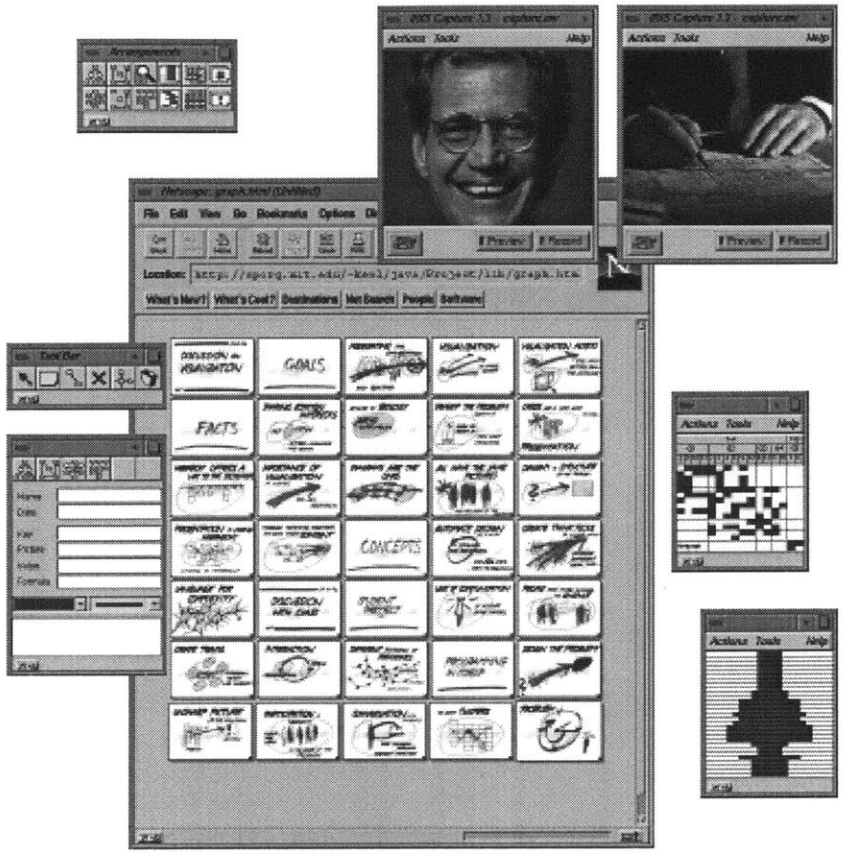
The attribute tool-bar allows for the insertion and modification of attributes which will be defined upon the creation of Card Wall objects. The tool-bar content will change based on the object selected on the Card Wall and based on the tool selected in the tool-menu. Figure 73 illustrates the attribute tool-bar as viewed when a link or the link tool is selected. A new link adapts all the attributes presently which are displayed at the time of the link's creation. The attributes of an existing link are modified by editing the values in the attribute tool-bar. The top-row in the attribute tool-bar displays all available link types. The selected button indicates the selected link type. The text insertion boxes below specify the name, date, keyword, picture path, value, and formula. The pull-down menus offer a selection of color and line thickness. The large insertion box on the button is reserved for explanatory text. The attribute menu always represents the attributes of the presently selected element. Attributes of cards and links are quickly viewed by selecting elements on the Card Wall. The card attribute tool-bar differs only slightly from the link attribute tool-bar. Some of the differences between the link attribute tool-bar and the card attribute tool-bar is that link types are replaced by the available card types and the link colors by priorities.

Layers


Layer Menu Figure 74

The layer tool-bar indicates in the first column the currently active layer, in the second column the visible layers, and in the third column the individual layer names. The layer tool-box displays the layer information of the presently selected element and allows for immediate changes. New layers can be created and layer names changed at anytime. More advanced layer managers are realized in programs such as Photoshop or Freehand. Layers can be rearranged to change the order that the layers are drawn to the screen. This is of help if cards on different layers are to overlap. A differentiation of cards on separate layers may also achieved through the assignment of transparencies or different colors to layers.

Figure 75 suggests a possible screen setting for a remote Card Wall session. The Card Wall is accessed through a web-browser. The tool-bars are located to the left. Visual and audible communication with other group members is maintained through video conferencing tools. Two windows display the output from the progression meter and the automatic group generator. The automatic group generator also indicates the people who are presently working on the Card Wall. The awareness of other people online is important to perceive the Card Wall not as a database but also as a place to meet and interact with other people. This allows for an increased interaction among participants and the occurrence of ad hoc behavior. This paper discussed many other Card Wall components that are not part of this illustration.



Possible Card Wall Components Figure 75

EPILOGUE

A *shared knowledge base* contains pieces of information that are accessible to many people within a design team. The *Card Wall* is an example of a shared knowledge base that visualizes pieces of information on individual cards. The *electronic Card Wall* is a computational version of the traditional Card Wall. It achieves distinct improvements over the original, such as the ability to visualize connections between cards. *Card connections* fall into five categories. There are network, hierarchy, process, group, or progression categories. The visualization of cards and connections among cards allows for easy transitions between the *concept elaboration, transformation, and formation* phase of the design process. Various *automatically generated card arrangements* allow for the recognition of relationships among cards. Certain types of relations are automatically recognized and visualized with the electronic Card Wall. The Card Wall contains only information about cards and the connections between cards. This makes it unnecessary to store a database of different spatial card arrangements. The connections between the cards contain sufficient information to retrieve past spatial arrangements. This permits remote participants to *individually view and arrange* the content of the shared knowledge base. User manipulations to the content of the shared knowledge base are realized through the introduction of a *voting system*. The different versions of the Card Wall process may be compared with a single view. An *intelligent separation of different Card Wall versions* is automatically computed and then visualized through the assignment of different levels of transparencies to cards and links. The electronic Card Wall *automatically defines groups of people* of common interest and understanding through the analysis of connections among cards which have been created by different people. The *progression of the Card Wall development* is visualized through the comparison of card connections and time stamps on cards. New cards are *considered to be additions, substitutions, variations, or combinations* of previous cards.

The electronic version of the Card Wall provides various enhancements over the manual manipulation of the Card Wall. The electronic Card Wall becomes applicable for many fields that focus *on idea development and group collaboration*. This is mainly achieved through the *advanced visualization of ideas and tools for the efficient analysis of relations among ideas*. Most of the proposed functionality is easy realizable, some requires further research. Some of the visualization techniques are based on intuitive assumptions. The refinement of such techniques requires the observation of people using the electronic Card Wall leading to a reevaluation of prior assumptions.

APPENDIX

The theoretical part of this research has been accompanied by the development of a prototype Card Wall. This appendix introduces some of the technical issues concerning the realization of this software.

The electronic Card Wall prototype is programmed in Java. Java is an object oriented programming language. The use of an object oriented programming language allows for the development of a very structured and expandable application. *Objects in an object orientated programming language are templates within the program which consist of other objects, functions, or attributes.* The Card Wall for example is an object that consists of card objects, arrangement functions, and dimensional information. The card objects themselves may contain link objects. An example of an object function which belongs to a card is a "move" function which modifies the coordinates within the card. An example of a link attribute is its current color or thickness (priority). Objects are sovereign with respect the rest of the objects within the program and can easily be exchanged between programs. This suggests a close relation between objects in an object oriented programming environment and cards on a Card Wall. For example, if a card is transferred to a different Card Wall, it will place itself into the same position as it was on the previous Card Wall and the links will connect the same cards as before. *Objects may inherit information from "parent" objects.* This means that cards may be built up from a number of parents who pass down or "inherit" functions or attributes. For example, a card has both a parent and a grandparent. The parent is called an Icon and the grandparent is called an ImageLabel. In addition to other attributes and functions, the card gets its ability to display a picture from the ImageLabel and its ability to be dragged with the mouse from the Icon.

Java applications are designed to enhance the functionality of web browsers such as Netscape or Microsoft Explorer. A compiled Java application runs on most available platforms and is — if accessed through a web browser — can be downloaded from a remote web server and executed on the clients machine. Programs which are downloaded from a network and are not trusted, pose a security risk because they may be able to destroy files on the client computer. Thus, for security reasons, data files may not be written or deleted on the client computer by the downloaded Java program. For this prototype, files may be modified or created on the server side only (where the Java program originated from). Keeping the data files on a central server is of advantage to the electronic Card Wall since remote Card Wall views can be updated after every change in Card Wall content. The updates are minimal in volume, since only changes in content but not in arrangements are transmitted. There is no need for fast network connections or servers since a few seconds delay in the update of a remote Card Wall view is of little concern for remote participants. The present version of the Card Wall prototype requires a user identification and password to download the program. After the program confirms the user name and password, it restores the last customized card arrangement. In the mean time, cards are added to the view in a designated location. The present version of the prototype only allows for changes of card arrangements by the system administrator. The electronic Card Wall prototype allows card elements such as pictures or text to be stored on any Internet server. This data decentralization allows users to efficiently connect external pieces of information with the Card Wall. The Card Wall becomes a visual hyper text language to connect digital documents (for example to visualize related Internet sites).

The electronic Card Wall prototype currently allow users to view, select, resize, move, and link cards. When the program starts up, it imports digitally stored pictures which have been scanned from real cardboard cards. When they are finished loading, the cards can be moved to any position on the screen by dragging with the mouse. The program redraws the entire Card Wall after every manipulation. If a Card Wall contains many cards and is viewed on a slow computer, this redrawing time might take a few seconds. This constant redrawing of the screen during a card's movement often causes flickering. To avoid from flickering, only a thin outline around the card is displayed as it is moved. There is presently little control over the order in which cards are redrawn. If two cards overlap, the card created last is displayed on top. The links are always displayed below the cards. Cards are easily linked by dragging a rubber band line from one card to another. The links do not yet support arrows. The next version of the electronic Card Wall prototype will take advantage of some newly developed Java libraries that control the order in which elements are drawn to the screen. This will permit the visualization of links on top of cards, and the ability of the user to define the order in which layers of cards are drawn to the screen. If an arrow is supposed to point to the card frame, its pixel shape and distance from the card center changes after every movement of the card. Those calculations increase the redraw time of the Card Wall and must be integrated into the program carefully. A later version of the prototype will also allow users to interactively change the card attributes such as keywords or pictures.

Only a few arrangement types are supported at this time. The circle arrangement, for example, places cards in an even distribution in a circle which is resized to fit the size of the window. The hierarchy arrangement organizes cards according to the rules which have been described previously in this paper. The realization of the basic functionality of the electronic Card Wall program has been a pre-requisite for most arrangement types. The programming effort for the realization of the basic functionality is most likely to exceed half of the total time for the realization of the electronic Card Wall prototype. The competition of the electronic Card Wall prototype is expected by summer 1997.

BIBLIOGRAPHY

Edith K. E. Ackermann — **Tools for Constructive Learning** - Rethinking Interactivity
MIT Media Lab, E&L Memo, October 1993, No. 15

Edith K. E. Ackermann — **Perspective-Taking and Object Construction**
Working Paper, 1994 MIT Media Lab

Arthur J. Deikman — **The Observing Self** - Mysticism and Psychotherapy
1982 Beacon Press

Douglas R. Hofstadter — **Gödel, Escher, Bach** - An Eternal Golden Braid
1979 Basic Books

Pattie Maes — **Modeling Adaptive Autonomous Agents**
Artificial Life Journal, 1994 MIT Press, Vol. 1, No. 1

Thomas W. Malone — **Tools for Inventing Organizations**
Towards a Handbook for Organizational Processes
Working Paper, 1992 MIT Center for Coordination Science

Thomas W. Malone — **Inventing the Organizations of the 21sh Century**
Control, Empowerment, and Information Technology
Working Paper, 1995 MIT SLOAN

Marvin L. Minsky — **The Society of Mind**
1988 Touchstone

Boris Mirkin — **Mathematical Classification and Clustering**
Nonconvex Optimization and its Applications
1996 Kluwer Academic Publishers

William J. Mitchell — **The Logic of Architecture**
1990 MIT Press

Jay F. Nunamaker — **Electronic Meeting System to Support Group Work**
Communications of the ACM, July 1991, Vol.34, No. 7, pp. 40-61

William M. Pena — **Problem Seeking** - An Architectural Programming Primer
1977 Cahners Books

Brian T. Pentland — **Useful Description of Organizational Processes**
Collecting Data for the Process Handbook
Working Paper, 1994 MIT Center for Coordination Science

Thomas J. Peters **Liberation Management**
Necessary Disorganization for the Nanosecond Nineties
1992 Excel

George Polya **How to Solve it** - A New Aspect of Mathematical Method
1945 Princeton University Press

Peter G. Rowe **Design Thinking**
1987 MIT Press

Donald A. Schon **The Reflective Practitioner** - How Professionals Think in Action
1983 Basic Books

Herbert A. Simon **The Sciences of the Artificial**
1981 MIT Press

Edward R. Tufte **Envisioning Information** - Narratives of Space and Time
1990 Graphics Press

Anthony Vidler **The Idea of Type** - The Transformation of the Academic Ideal

Mitchell M. Waldrop **Complexity** - The emerging Science at the Edge of Order and Chaos
1992 Touchstone

Patrick H. Winston **Artificial Intelligence**
1992 Addison Wesley

Alfred N. Whitehead **The Aims of Education and Other Essays**
1957 Macmillan Company, New York

Edward Yourdon **Modern Structured Analysis**
Tools of Structured Analysis and Dataflow Diagrams
1989 Englewood Clifs

All illustrations are by the author unless otherwise noted.